Accelerated Gossip Algorithms for Distributed Computation

Ming Cao[1] Daniel A. Spielman[2] Edmund M. Yeh[1]
[1]Department of Electrical Engineering and [2]Department of Computer Science
Yale University, New Haven, CT 06520, USA
{m.cao, daniel.spielman, edmund.yeh}@yale.edu

***Abstract*— We introduce a technique for accelerating the gossip algorithm of Boyd et. al. (INFOCOM 2005) for distributed averaging in a network. By employing memory in the form of a small shift-register in the computation at each node, we can speed up the algorithm's convergence by a factor of 10. Our accelerated algorithm is inspired by the observation that the original gossip algorithm is analogous to the power method in Numerical Analysis, which can be accelerated by a shift-register based recurrence.**

I. INTRODUCTION

Recently, there has been rapidly growing interest in gossip-type algorithms for applications in large-scale wireless sensor networks [1], [2], [3], [4]. Gossip algorithms in sensors networks are special network consensus algorithms in which all sensors are required to agree on the same value, e.g. the average of all the sensors' initial measurements. The average value is of special importance since it is crucial for designing many distributed sensor fusion algorithms [4]. The authors of [2] study a gossip algorithm where each sensor updates its value from time to time to a new value equal to the average of its current value and the value of one randomly chosen neighbor. This algorithm has a number of important advantages. First, it is fully distributed and does not require centralized coordination. Second, it is asynchronous and does not require synchronized clocks at the sensors. Third, the algorithm is easy to implement and each node performs only a simple computation at each step. Fourth, the algorithm is relatively fault-tolerant in the sense that it can operate even with a high probability of link failures in sensor networks placed in hazardous environments. In spite of these advantages, however, one critical drawback of the algorithm in [2] is its relatively low convergence rate, which places an undesirably high demand on the communication and computational resources in the network. For example, in a random geometric graph of bounded expected degree, even the optimized version of the gossip algorithm in [2] requires $\Theta(n^2)$ iterations for convergence to an error level of $\varepsilon = n^{-\alpha}$ for $\alpha > 0$, where n is the number of nodes in the network.

The work of Cao was supported in part, by grants from the U.S. Army Research Office and the U.S. National Science Foundation and by a gift from the Xerox Corporation.

The work of Spielman was supported partially by NSF grant CCR-0324914.

The work of Yeh was supported in part, by Army Research Office (ARO) Young Investigator Program (YIP) grant DAAD19-03-1-0229 and grant W911NF-05-1-0383, National Science Foundation (NSF) grant CCR-0313183, and Air Force Office of Scientific Research (AFOSR) grant FA9550-06-1-0135.

A number of recent papers have attempted to address the issue of slow convergence in gossip algorithms. In [4], a consensus propagation algorithm based on belief propagation is used to improve the convergence. In [5], a probabilistic counting algorithm is used. These algorithms have been analyzed for specific graphs. Their efficacy in more general networks has not yet been fully demonstrated. In [6], a modified gossip algorithm is proposed to accelerate the convergence rate by exploiting each sensor's location information. This algorithm, however, assumes the existence of a network localization protocol which itself demands communication and computational resources.

In this paper, we address the issue of accelerating the convergence process of the original gossip algorithm using a completely different approach. The main idea is to employ more *memory* in the gossip algorithm to substantially improve the convergence rate. We assume each sensor is provided with a small shift-register storing its values in the finite past. When a sensor updates, the values stored in its shift-register as well as the current value of its (randomly) chosen neighbor will be used in the computation. Our accelerated gossip algorithm does not increase the communication cost and only slightly increases the computation at each step. At the same time, however, the algorithm *increases the convergence rate by a factor of 10.* Our accelerated algorithm is inspired by the observation that the original gossip algorithm is analogous to the power method in Numerical Analysis, which can be accelerated by a shift-register based recurrence [7].

II. ACCELERATED GOSSIP ALGORITHM

We first review the original gossip algorithm studied in [2]. Consider a senor network consisting of n nodes, labelled 1 through n. For each $i \in \{1,\ldots,n\}$, let $x_i(t)$ denote the value of node i at time t. Each node has an initial value at time $t = 0$. At any time-step $t \in \{1,2,\ldots\}$, each node has an equal probability $\frac{1}{n}$ of being activated to update its value. At time t, if node i is activated, then with probability p_{ij} node i chooses its neighbor j to update their values together to the mean of their current values:

$$\begin{cases} x_i(t+1) = \frac{1}{2}(x_i(t) + x_j(t)) \\ x_j(t+1) = \frac{1}{2}(x_i(t) + x_j(t)) \end{cases} \tag{1}$$

The values of all the remaining nodes remain the same:

$$x_k(t+1) = x_k(t), \qquad \text{for all } k \neq i \text{ and } k \neq j \tag{2}$$

It is possible to put the iterative equations (1) and (2) for the n-node system into state form. Towards this end, define the system state as:

$$x(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t)]' \tag{3}$$

Then the system evolves according to

$$x(t+1) = W(t)x(t), \qquad t = 1,2,\ldots \tag{4}$$

where with probability $\frac{1}{n}p_{ij}$,

$$W(t) = W_{ij} \triangleq I - \frac{(e_i - e_j)(e_i - e_j)'}{2} \tag{5}$$

Here e_i is the i-th n-dimensional elementary column vector, I is the n-dimensional identity matrix, and v' denotes the transpose of vector v.

The conditions under which the system (4) converges are given with respect to the expectation of the matrix $W(t)$. Let

$$V \triangleq E[W_{ij}] = \sum_{i,j} \frac{1}{n} p_{ij} W_{ij} \tag{6}$$

Since the W_{ij}'s are doubly stochastic matrices [8], V is also a doubly stochastic matrix. The results in [2] claim that using the gossip algorithm (1) and (2), each node's value converges to the average value $x_{\text{ave}} = \frac{1}{n}\sum_i x_i(0)$ if V has 1 as a simple eigenvalue and all the remaining $n-1$ eigenvalues are strictly less than 1 in magnitude. Furthermore, the convergence rate is governed by V's second largest eigenvalue. It has also been shown in [2] that if a node can only communicate with a neighbor within a small range r, then this process converges slowly.

Since the original gossip algorithm in its state form (4) is analogous to the power method in Numerical Analysis, we are inspired to use memory in the form of shift-registers to accelerate the original gossip algorithm. This is similar to what has been done to successfully speed up the power method in the past few decades [7]. We will not change the pair of nodes participating in an update at time $t = 1,2,\ldots$ We will only change how a pair of nodes update their values once they decide to update together. For sensor i, $i \in \{1,\ldots,n\}$, let x_{ir} denote the value stored in its rth register. We first consider the case where each sensor is provided with two registers, the first of which stores the sensor's current value and the second of which stores the sensor's value before the latest update. If at time $t \in \{1,2,\ldots,\}$, nodes i and j are the pair to update their current values together, then we propose the following accelerated gossip algorithm:

$$\begin{cases} x_{i1}(t+1) = \omega(\frac{1}{2}x_{i1}(t) + \frac{1}{2}x_{j1}(t)) + (1-\omega)x_{i2}(t) \\ x_{i2}(t+1) = x_{i1}(t) \\ x_{j1}(t+1) = \omega(\frac{1}{2}x_{i1}(t) + \frac{1}{2}x_{j1}(t)) + (1-\omega)x_{j2}(t) \\ x_{j2}(t+1) = x_{j1}(t) \end{cases} \tag{7}$$

where $1 \le \omega < 2$ is a constant. The values of the registers of all the other nodes remain the same: for all $k \ne i$ and $k \ne j$,

$$\begin{cases} x_{k1}(t+1) = x_{k1}(t) \\ x_{k2}(t+1) = x_{k2}(t) \end{cases} \tag{8}$$

n	ρ	# edges	min degree	max degree
10000	.02	124,000	1	30
10000	.04	486,000	13	81
20000	.0141	246,000	1	28

TABLE I

We can generalize the above update rules (7) and (8) to the case where each sensor is installed with $m > 2$ registers. Again suppose at time $t \in \{1,2,\ldots,\}$, nodes i and j are the pair to update their current values together. Then the accelerated gossip algorithm using m registers at each node is the following:

$$\begin{cases} x_{i1}(t+1) = \omega_1\left(\frac{1}{2}x_{i1}(t) + \frac{1}{2}x_{j1}(t)\right) + \sum_{r=2}^{m} w_r x_{ir}(t) \\ x_{ir}(t+1) = x_{i(r-1)}(t), \qquad r = 2,\ldots,m \\ x_{j1}(t+1) = \omega_1\left(\frac{1}{2}x_{i1}(t) + \frac{1}{2}x_{j1}(t)\right) + \sum_{r=2}^{m} w_r x_{jr}(t) \\ x_{jr}(t+1) = x_{j(r-1)}(t), \qquad r = 2,\ldots,m \end{cases} \tag{9}$$

where ω_r, $r = 1,\ldots,m$, are constants satisfying $\sum_{r=1}^{m} \omega_r = 1$. The values of the registers of all the other nodes remain the same: for all $k \ne i$ and $k \ne j$,

$$x_{kr}(t+1) = x_{kr}(t), \qquad r = 1,\ldots,m \tag{10}$$

In the following section, we will evaluate the proposed accelerated gossip algorithm by carrying out various experiments on random geometric graphs, which are commonly used to model wireless sensor networks. The experimental results will provide compelling evidence for the superior convergence rate performance of the accelerated gossip algorithm.

In Section IV, we prove that when the accelerated algorithm converges it must converge to the correct value.

III. Experimental Results

In all of our experiments, we used random geometric graphs generated by choosing n points at random in the unit square, and then placing an edge between each pair of points at distance less than ρ from each other. If the graph was disconnected, we used the largest connected component. In every experiment, the largest connected component missed at most 3 vertices. Table I gives typical statistics of the graphs we used.

We ran our experiments with four different distributions of initial values.

1. Normal: each initial value is chosen independently at random from the Normal distribution.
2. Normal about x-coord: the initial value of each point is independently chosen from the Normal distribution of variance 1 whose mean is the x-coordinate of the point.
3. 0/1 at x-coord: each initial value is independently chosen to be 0/1, with the probability of being 1 equal to the x-coordinate.

D2	1.4386, -0.4386
D4	1.286, 0, 0, -0.286
D8	1.128, 0, 0, 0, 0, 0, 0, -0.128
X4	1.280, 0.08, -0.06, -0.30

TABLE II

4. x-coord: the x-coordinates of the points are used as the initial values.

We ran experiments with four different configurations of shift registers. The first three, denoted D2, D4 and D8 consisted of 2, 4 and 8 registers. In these, only the first and last register were used to compute the new value of the first register. In the other, denoted X4, all of the registers were used to compute the new value of the first register. The vectors of coefficients for each were shown in Table II.

In each experiment, we measured convergence by the average of the square of the difference between the current value of a node and x_{ave}, the average to be computed.

In Figures 1, 2, 3 and 4 we give the results of these experiments for the graphs generated on 10,000 nodes with $\rho = .02$. In each case, we have performed 100 experiments, and plotted the geometric mean of the data. In Figures 5, 6, 7, 8, we plot the geometric means, as well as the maxima and minima of the data.

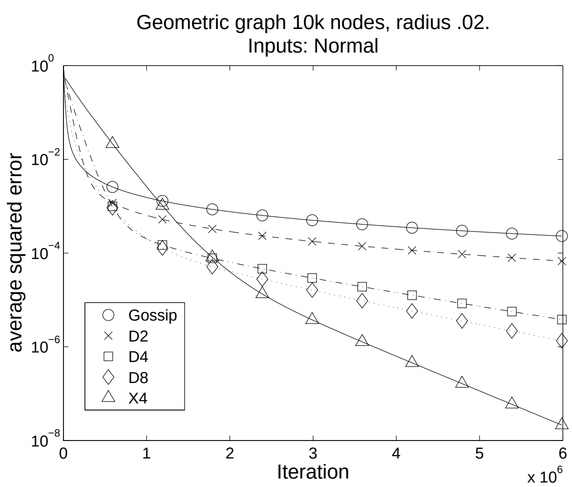

Fig. 1. Geometric means of 100 trials

Figures 9, 10, 11 and 12 show how the convergence rates change as we increase the radius of the geometric graphs to .04. As before, each curve is the geometric mean of 100 experiments.

The curves for X4 and D8 in Figure 11 and 12 are in fact wavy, and were tightly concentrated about the geometric means plotted here.

We now examine how the convergence rate changes as we increase the number of vertices. We ran the same set of experiments on random geometric graphs with 20,000 vertices, connecting vertices within distance at most 0.0141. This choice of distance parameter makes the average-degree of these graphs comparable to that of the graphs on 10,000

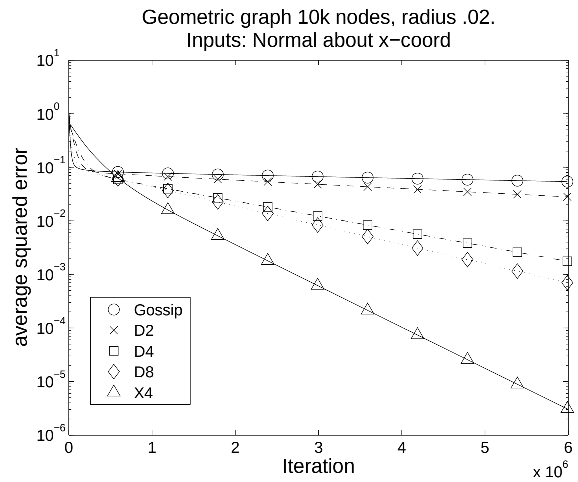

Fig. 2. Geometric means of 100 trials

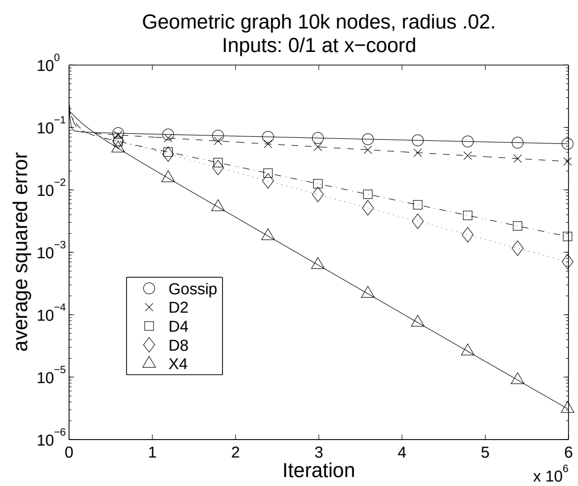

Fig. 3. Geometric means of 100 trials

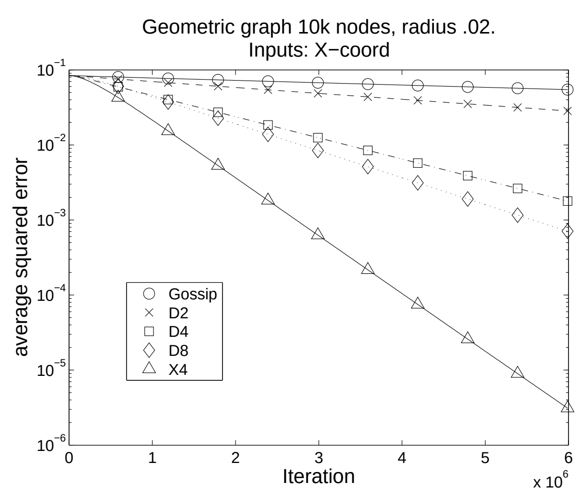

Fig. 4. Geometric means of 100 trials

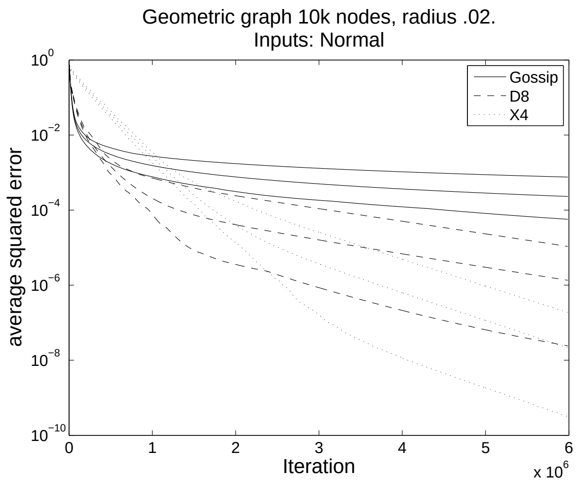

Fig. 5. Minima, geometric means and maxima of 100 trials

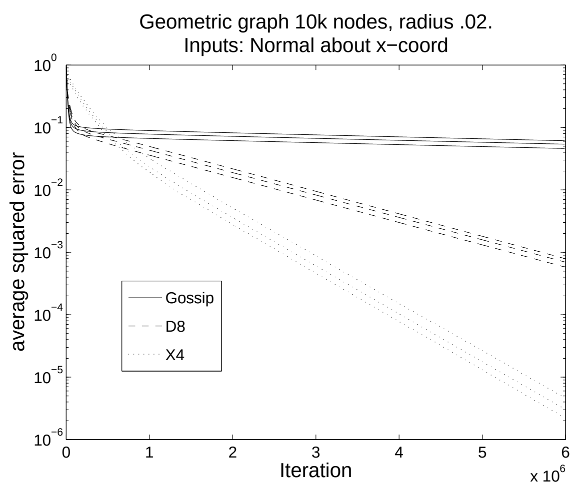

Fig. 6. Minima, geometric means and maxima of 100 trials

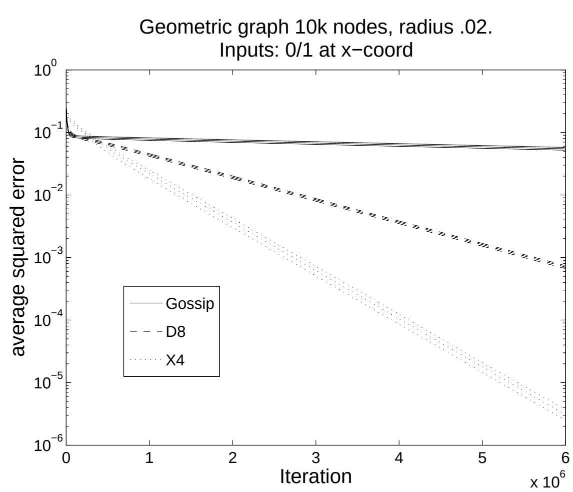

Fig. 7. Minima, geometric means and maxima of 100 trials

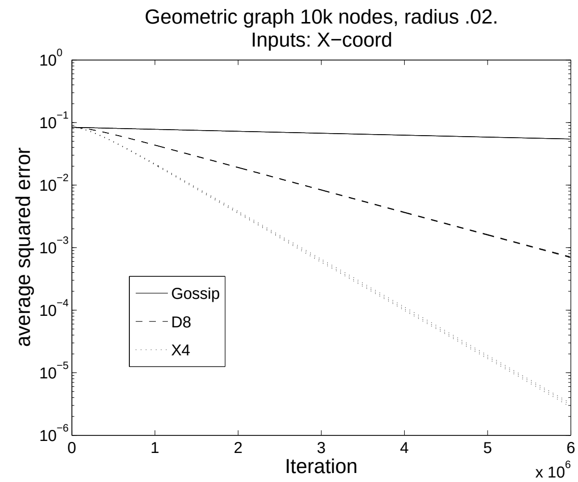

Fig. 8. Minima, geometric means and maxima of 100 trials

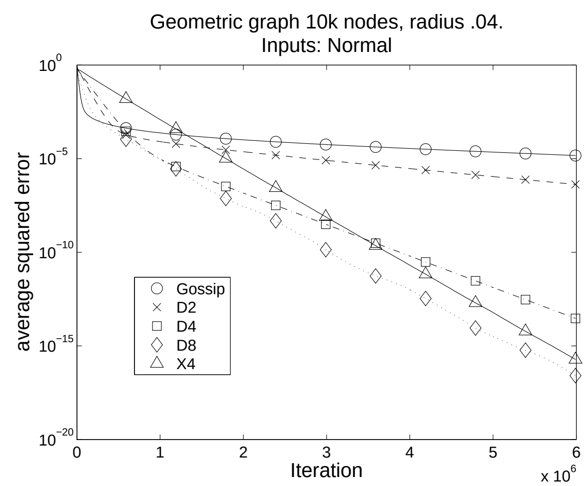

Fig. 9. Geometric means of 100 trials

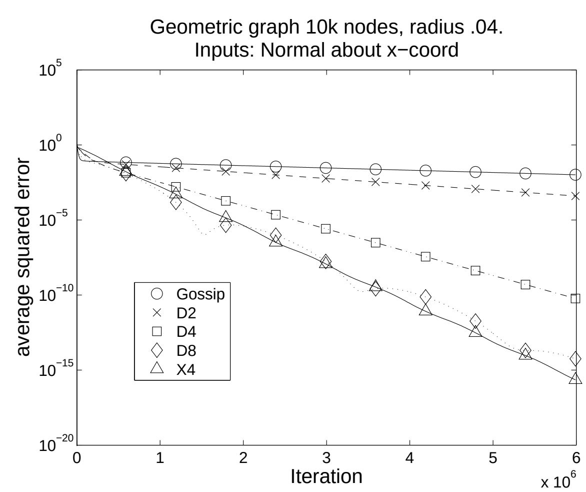

Fig. 10. Geometric means of 100 trials

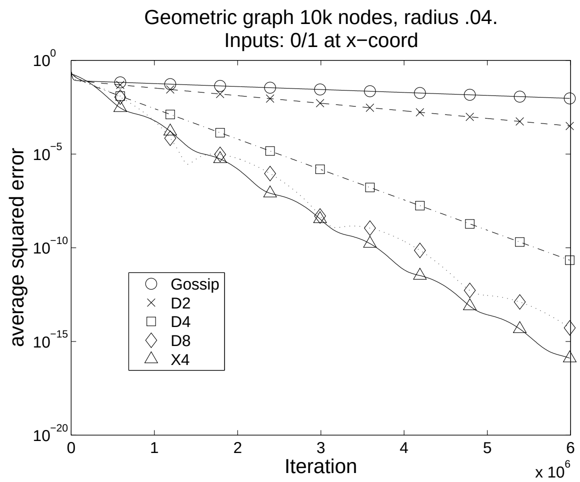

Fig. 11. Geometric means of 100 trials

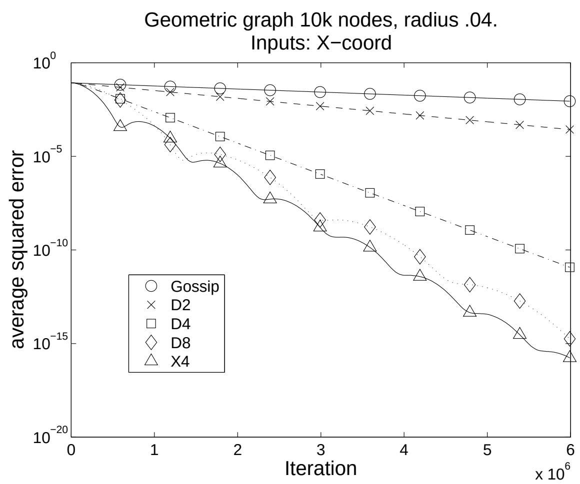

Fig. 12. Geometric means of 100 trials

vertices and distance parameter 0.02. We ran these experiments for four times as many iterations as the experiments on 10,000 vertices so that they would converge similiarly. The results of these experiments appear in Figures 13, 14, 15 and 16.

In all the experiments in which the input was not the x-coordinate, the original gossip algorithm converged faster initially, with D8 trailing only slightly behind. In all cases, the asymptotic performance of the original gossip algorithm was dominated by the accelerated algorithms. In particular, for all $\varepsilon < 0.5$, the ε-averaging times, as defined by [2], was always lower for the accelerated algorithms.

Table III gives the slopes, times 10^9, of the lines that best-fit the last half of each of the curves in the tables plotted so far. Note that a larger absolute value indicates faster convergence.

To get an idea of how the algorithm's behavior changes as n grows, we divide the first four lines in Table III by the last four, obtaining Table IV.

The reader will observe that almost all the values in this table are a little more than 4. This is consistent with the

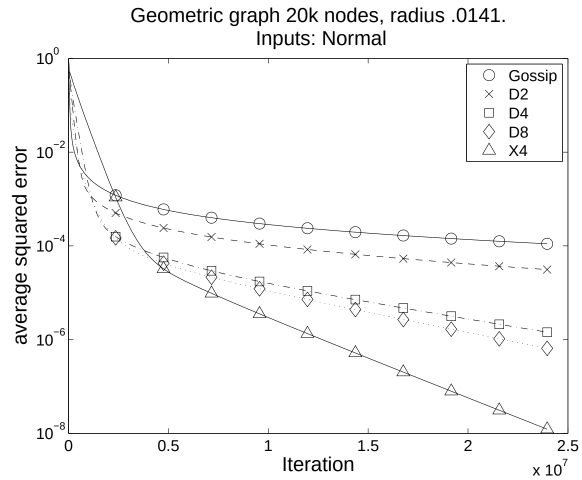

Fig. 13. Geometric means of 100 trials

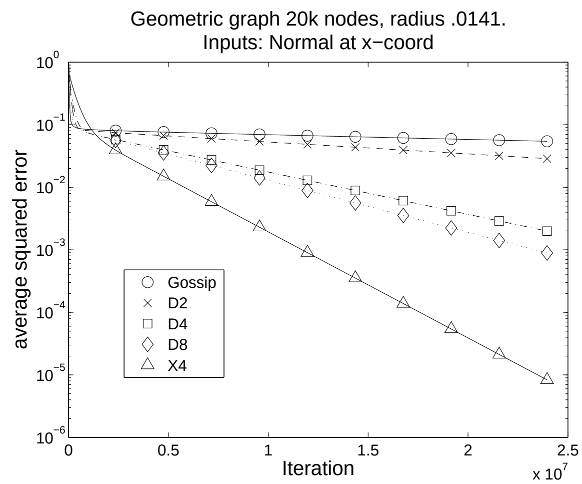

Fig. 14. Geometric means of 100 trials

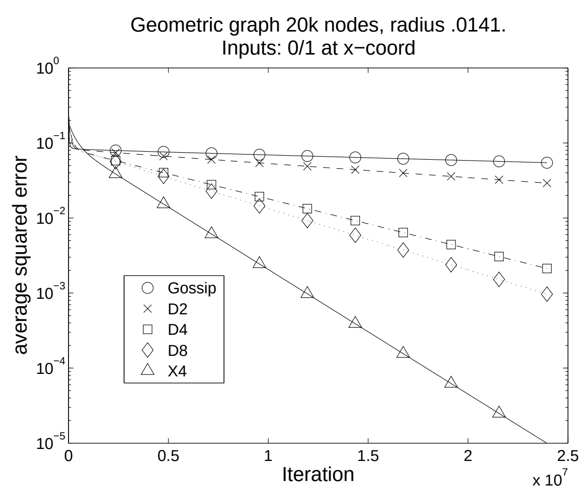

Fig. 15. Geometric means of 100 trials

Figure	n	ρ	Inputs	Gossip	D2	D4	D8	X4
1	10,000	0.02	Normal	-254.79	-317.57	-678.54	-826.54	-1712.8
2	10,000	0.02	Normal at X	-71.742	-180.22	-648.88	-826.14	-1767
3	10,000	0.02	0/1 at X	-70.835	-179.88	-648.79	-826.08	-1769.5
4	10,000	0.02	X coord	-70.709	-179.83	-648.92	-825.85	-1769.9
9	10,000	0.04	Normal	-446.97	-985.61	-3859.8	-5106.4	-5844
10	10,000	0.04	Normal at X	-349.17	-895.51	-3567	-4931.6	-5838.8
11	10,000	0.04	0/1 at X	-364.07	-933.39	-3725.6	-4783.6	-5769.2
12	10,000	0.04	X coord	-374.56	-958.37	-3819.4	-5005.5	-5617.4
13	20,000	0.0141	Normal	-62.602	-81.523	-168.53	-200.08	-391.04
14	20,000	0.0141	Normal at X	-17.329	-43.734	-155.89	-191.87	-390.44
15	20,000	0.0141	0/1 at X	-16.973	-42.963	-153.02	-188.15	-382.15
16	20,000	0.0141	X coord	-16.771	-42.557	-151.72	-186.61	-379.09

TABLE III

Figure	Gossip	D2	D4	D8	X4
1 / 13	4.07	3.8955	4.0264	4.1311	4.3803
2 / 14	4.14	4.1209	4.1625	4.3057	4.5256
3 /15	4.1734	4.1869	4.2399	4.3905	4.6304
4 / 16	4.2161	4.2256	4.277	4.4255	4.6689

TABLE IV

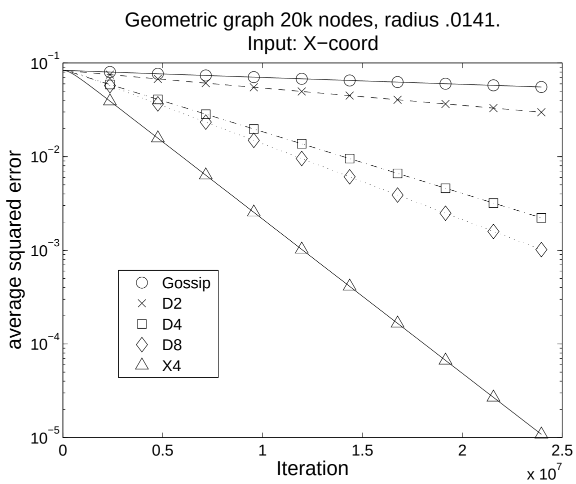

Fig. 16. Geometric means of 100 trials

analysis of [2], which indicates that the convergence rate of the Gossip process should be proportional to the logarithm of the second-largest eigenvalue of the matrix V. In the case of the graphs on 20,000 vertices, this quantity was $9.18 \cdot 10^{-9}$, while for the graph on 10,000 vertices, this quantity was $3.61 \cdot 10^{-8}$, and the ratio of these quantities was 3.93. In general, for geometric graphs whose radii are set to keep the average degree fixed, one would expect the logarithm of this eigenvalue to grow like the square of the number of vertices.

Table V gives, for each experiment, the approximate iteration number at which each curve achieves the value that the gossip algorithm achieved at the last iteration. The reason these iteration numbers are only approximate is that the average squared error was only computed at 600 different times during each simulation.

To see exactly how much less time it took the shift-register based algorithms to converge, we divide all of the columns of Table V by the first obtaining Table VI.

IV. Analysis

In this section, we show that if the accelerated gossip algorithm converges, it always converge to the correct value. One feature of the original gossip algorithm is that the sum of all the sensors' values is preserved during the system evolution. The accelerated gossip algorithm has a similar feature of preserving a scaled sum. We only give a proof of this conclusion for the 2-register case. The general m-register case can be proved using similar ideas.

To write the iterative equations (7) and (8) into their state form, define the enlarged system state as:

$$\begin{aligned} z(t) &= [x_{11}(t) \quad x_{21}(t) \quad \cdots \quad x_{n1}(t) \\ &\qquad x_{12}(t) \quad x_{22}(t) \quad \cdots \quad x_{n2}(t)]' \end{aligned} \tag{11}$$

Then

$$z(t+1) = \Gamma(t) z(t), \qquad t = 1,2,\ldots \tag{12}$$

where with probability $\frac{1}{n} p_{ij}$,

$$\Gamma(t) = A_{ij} \triangleq \begin{bmatrix} \sum_{k\neq i,j} e_k e_k' + \frac{\omega}{2}(e_i+e_j)(e_i+e_j)' & (1-\omega)(e_i e_i' + e_j e_j') \\ e_i e_i' + e_j e_j' & \sum_{k\neq i,j} e_k e_k' \end{bmatrix} \tag{13}$$

Theorem 1: During the evolution of system (12), the scaled sum $\frac{1}{2-\omega}\sum_{i=1}^n x_{i1}(t) + \frac{1-\omega}{2-\omega}\sum_{i=1}^n x_{i2}(t)$ is preserved for $t = 1,2,\ldots$, i.e.

$$\begin{aligned} & \frac{1}{2-\omega}\sum_{i=1}^n x_{i1}(t) + \frac{1-\omega}{2-\omega}\sum_{i=1}^n x_{i2}(t) \\ = & \frac{1}{2-\omega}\sum_{i=1}^n x_{i1}(0) + \frac{1-\omega}{2-\omega}\sum_{i=1}^n x_{i2}(0) \\ & t = 1,2,\ldots \end{aligned} \tag{14}$$

Figure	n	ρ	Inputs	Gossip	D2	D4	D8	X4
1	10,000	0.02	Normal	5,988,802	2,409,518	939,812	969,806	1,529,694
2	10,000	0.02	Normal at X	5,988,802	2,369,526	729,854	729,854	649,870
3	10,000	0.02	0/1 at X	5,988,802	2,359,528	729,854	719,856	509,898
4	10,000	0.02	X coord	5,988,802	2,359,528	729,854	719,856	439,912
9	10,000	0.04	Normal	5,990,000	2,430,000	940,000	920,000	1,730,000
10	10,000	0.04	Normal at X	5,990,000	2,360,000	680,000	630,000	670,000
11	10,000	0.04	0/1 at X	5,990,000	2,360,000	670,000	620,000	460,000
12	10,000	0.04	X coord	5,990,000	2,360,000	670,000	620,000	380,000
13	20,000	0.0141	Normal	23,960,000	9,560,000	3,000,000	2,720,000	3,560,000
14	20,000	0.0141	Normal at X	23,958,802	9,439,528	2,799,860	2,559,872	1,719,914
15	20,000	0.0141	0/1 at X	23,952,812	9,397,180	2,799,160	2,559,232	1,519,544
16	20,000	0.0141	X coord	23,958,802	9,399,530	2,799,860	2,559,872	1,439,928

TABLE V

Figure	n	ρ	Inputs	Gossip	D2	D4	D8	X4
1	10,000	0.02	Normal	1	0.40234	0.15693	0.16194	0.25543
2	10,000	0.02	Normal at X	1	0.39566	0.12187	0.12187	0.10851
3	10,000	0.02	0/1 at X	1	0.39399	0.12187	0.1202	0.085142
4	10,000	0.02	X coord	1	0.39399	0.12187	0.1202	0.073456
9	10,000	0.04	Normal	1	0.40568	0.15693	0.15359	0.28881
10	10,000	0.04	Normal at X	1	0.39399	0.11352	0.10518	0.11185
11	10,000	0.04	0/1 at X	1	0.39399	0.11185	0.10351	0.076795
12	10,000	0.04	X coord	1	0.39399	0.11185	0.10351	0.063439
13	20,000	0.0141	Normal	1	0.399	0.12521	0.11352	0.14858
14	20,000	0.0141	Normal at X	1	0.39399	0.11686	0.10684	0.071786
15	20,000	0.0141	0/1 at X	1	0.39232	0.11686	0.10684	0.063439
16	20,000	0.0141	X coord	1	0.39232	0.11686	0.10684	0.0601

TABLE VI

To prove Theorem 1, we need the following result:

Lemma 1: Let $\mathbf{1}$ be the n-dimensional all ones column vector. Then $[\frac{1}{2-\omega}\mathbf{1}' \quad \frac{1-\omega}{2-\omega}\mathbf{1}']$ is a left eigenvector associated with eigenvalue 1 of the matrix A_{ij} defined in (13) for all $i,j \in \{1,\dots,n\}$ and $i \neq j$.

Proof of Lemma 1: Let $v = [\frac{1}{2-\omega}\mathbf{1}' \ \frac{1-\omega}{2-\omega}\mathbf{1}']A_{ij}$. Then v is a $2n$-dimensional row vector. One can check that for all $i,j \in \{1,\dots,n\}$ and $i \neq j$,

(a) The ith entry of v is $\frac{1}{2-\omega}(\frac{\omega}{2}+\frac{\omega}{2})+\frac{1-\omega}{2-\omega}=\frac{1}{2-\omega}$;
(b) The jth entry of v is $\frac{1}{2-\omega}(\frac{\omega}{2}+\frac{\omega}{2})+\frac{1-\omega}{2-\omega}=\frac{1}{2-\omega}$;
(c) For $1 \le k \le n$, $k \neq i$ and $k \neq j$, the kth entry of v is $= \frac{1}{2-\omega}$;
(d) The $(n+i)$th entry of v is $\frac{1}{2-\omega}(1-\omega)=\frac{1-\omega}{2-\omega}$;
(e) The $(n+j)$th entry of v is $\frac{1}{2-\omega}(1-\omega)=\frac{1-\omega}{2-\omega}$;
(f) For $1 \le k \le n$, $k \neq i$ and $k \neq j$, the $(n+k)$th entry of v is $\frac{1-\omega}{2-\omega}$;

Combining (a)-(f), we have $v = [\frac{1}{2-\omega}\mathbf{1}' \ \frac{1-\omega}{2-\omega}\mathbf{1}']$ which implies the conclusion. □

Proof of Theorem 1: We can rewrite the scaled sum into a vector product as follows:

$$
\begin{aligned}
& \frac{1}{2-\omega}\sum_{i=1}^{n} x_{i1}(t)+\frac{1-\omega}{2-\omega}\sum_{i=1}^{n} x_{i2}(t) \\
= & \left[\frac{1}{2-\omega}\mathbf{1}' \quad \frac{1-\omega}{2-\omega}\mathbf{1}'\right] z(t) \\
& \text{(by the definition of } z(t)) \\
= & \left[\frac{1}{2-\omega}\mathbf{1}' \quad \frac{1-\omega}{2-\omega}\mathbf{1}'\right] \Gamma(t-1)z(t-1) \\
& \text{(by equation (12))} \\
= & \left[\frac{1}{2-\omega}\mathbf{1}' \quad \frac{1-\omega}{2-\omega}\mathbf{1}'\right] z(t-1) \\
& \text{(by Lemma 1)} \\
= & \cdots \\
= & \left[\frac{1}{2-\omega}\mathbf{1}' \quad \frac{1-\omega}{2-\omega}\mathbf{1}'\right] z(0) \\
& \text{(by repeated use of Lemma 1)} \\
= & \frac{1}{2-\omega}\sum_{i=1}^{n} x_{i1}(0)+\frac{1-\omega}{2-\omega}\sum_{i=1}^{n} x_{i2}(0)
\end{aligned}
$$

□

The direct consequence of the preservation of the scaled sum is that each node's value will converge to the desired value $\frac{1}{n}\sum_{i=1}^{n} x_{i1}(0)$ if all nodes' values asymptotically converge to the same one and each node's two registers are initialized so that $x_{i1}(0) = x_{i2}(0)$ for all $i = 1,\dots,n$.

V. Conclusion and Future Work

From the experiments presented here, it is clear that gossip algorithms for averaging can be greatly accelerated by the use of shift registers. We believe that it is possible to design coefficient vectors that give even better performance. Open questions suggested by this paper include:

1. Is it possible to synthetically analyze the performance of these accelerated algorithms, as [2] did for their gossip algorithm?
2. Just how much speed-up can one achieve through the use of shift-registers?
3. Through the use of shift registers, is it possible to make algorithms whose convergence time on geometric

graphs is proportional to the diameter of these graphs, rather than to the number of nodes in these graphs?

REFERENCES

[1] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. of ACM Symposium on Theory of Computing*, 2004.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. of the 24th Conference of the IEEE Communications Society (INFOCOM)*, 2005.

[3] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: Distributed randomized algorithms and analysis," in *Proc. of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[4] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Transactions on Information Theory*. To appear.

[5] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in *Proc. of the 25th annual ACM symposium on principles of distributed computing*, 2005.

[6] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient aggregation for sensor networks," in *Proc. of the 5th International Symposium on Information Processing in Sensor Networks (IPSN)*, 2006.

[7] D. M. Young, "Second-degree iterative methods for the solution of large linear systems," *Journal of Approximation Theory*, vol. 5, pp. 137–148, 1972.

[8] R. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1985.