

Cluster-based Distributed Consensus

Wenjun Li, *Member* and Huaiyu Dai*, *Member*

Abstract

In this paper, we incorporate clustering techniques into distributed consensus algorithms for faster convergence and better energy efficiency. Together with a simple distributed clustering algorithm, we design cluster-based distributed consensus algorithms in forms of both fixed linear iteration and randomized gossip. The time complexity of the proposed algorithms is presented in terms of metrics of the original and induced graphs, through which the advantage of clustering is revealed. Our cluster-based algorithms are also shown to achieve an $\Omega(\log n)$ gain in message complexity over the standard ones.

Index Terms

Clustering, Distributed computing, Distributed consensus, Sensor networks.

I. INTRODUCTION

The distributed consensus problem where nodes try to reach consensus on the average value¹ through iterative local information exchange has been vigorously investigated recently (see [1], [2] and references therein). In contrast to traditional centralized schemes, such distributed algorithms scale well as the network grows, and exhibit robustness to node and link failures. Distributed consensus is typically realized in a linear iteration framework with a graph conformant

This research was supported in part by the National Science Foundation under Grant CCF-0515164 and CNS-0721815.

W. Li is with Qualcomm Inc, San Diego, CA 92126 (e-mail: wenjunl@qualcomm.com). The work was done when she was with NC State University.

H. Dai is with the ECE department of NC State University, Raleigh, NC 27695 (e-mail: huaiyu_dai@ncsu.edu).

¹With modification, computation of weighted sums, linear synopses, and histograms can also be addressed.

weight matrix, either fixed or time varying. The optimal fixed iteration is studied in [1]. In [2], a randomized gossip algorithm² is proposed, which is more robust to network dynamism and can be implemented in an asynchronous manner. Concerns on this set of fully-distributed algorithms, however, include the incurred delay, and sometimes the associated energy consumption as well.

In this paper, taking a hybrid approach, we incorporate clustering techniques into distributed consensus algorithms to achieve faster convergence as well as reduce communication and computational complexity. This is motivated by two main observations. First, clustering usually improves the connectivity of a network, thus facilitate the convergence of distributed algorithms. Second, the broadcast nature of the wireless medium can be exploited for better energy efficiency. After introducing a simple and effective distributed clustering algorithm, we design cluster-based fixed iteration and randomized gossip consensus algorithms, for which computations are carried out by cluster-heads and some gateway nodes, and nodes in a cluster maintain the same value at any time from the broadcast of the cluster-head, effectively behaving as a single entity. The time complexity of the proposed algorithms is presented in terms of metrics of the original and induced graphs, through which the advantage of clustering is revealed. Our cluster-based algorithms are also shown to achieve an $\Omega(\log n)^3$ gain in message complexity⁴ over the standard ones.

The rest of this paper is organized as follows. In Section II, we formulate the problem, describe the distributed clustering algorithm, and lay the foundation for analysis. In Section III, we propose cluster-based distributed averaging algorithms with fixed iteration and randomized gossip, whose performance is analyzed and verified with simulation results. Section IV concludes this paper.

²In gossip algorithms, a node can communicate with only one neighbor at any time instant.

³Let $f(n)$ and $g(n)$ be nonnegative functions for $n \geq 0$. We say $f(n) = O(g(n))$ and $g(n) = \Omega(f(n))$ if there exists some k and $c > 0$, such that $f(n) \leq cg(n)$ for $n \geq k$.

⁴When the distances between adjacent nodes are small, the energy cost is well modeled by the message complexity.

II. PROBLEM FORMULATION AND CLUSTERING

A. Problem Formulation

Consider a network represented by a connected graph $G = (V, E)$, where the vertex set V contains n nodes and E is the edge set. Let $\mathbf{x}(0) = [x_1(0), \dots, x_n(0)]^T$ be the initial values of nodes, and $x_{\text{ave}} = \frac{1}{n} \sum_{i=1}^n x_i$ the average. The goal is to compute x_{ave} in a distributed and robust fashion.

We will mainly use the geometric random graph $G(n, r(n))$ [3] to model a wireless network in our analysis, where n nodes are uniformly and independently distributed on a unit square, and $r(n)$ is the common transmission range of all nodes. It is known that the choice of $r(n) \geq \sqrt{\frac{2 \log n}{n}}$ is required to ensure the graph is connected with high probability [3].

B. Distributed Clustering

We assume that clusters are formed in such that 1) each node belongs to one and only one cluster and 2) in each cluster, there is a cluster-head which is adjacent to all the remaining nodes. Two clusters are said to be neighbors if there is a direct link joining them. If more than one such links exist, one of them is activated. The end nodes of active links are called gateway nodes. The set of cluster-heads by our assumption is a dominating set, i.e., a subset of nodes which are at most 1 hop away from any node. The minimum dominating set (MDS) problem is NP-hard [4]. As the complexity of MDS approximation is not justified in large dynamic networks, we introduce a simple non-iterative decentralized clustering algorithm as follows.

We assume each node i has an initial seed s_i which is unique within its neighborhood. This can be realized through, e.g., drawing a random number from a common pool, or simply using nodes' IDs. From time 0, each node i starts a timer with length $t_i = s_i$, which is decremented by 1 at each time instant as long as it is greater than 0. If node i 's timer expires, it becomes a cluster-head, and broadcasts a "cluster_initialize" message to all its neighbors. Each of its neighbors with a timer greater than 0 signals its intention to join the cluster by replying with a "cluster_join" message,

and also sets the timer to 0. If a node receives more than one “cluster_initialize” messages at the same time, it randomly chooses one cluster-head. At the end, clusters are formed such that every node belongs to one and only one cluster. Note that the uniqueness of seeds within the neighborhood ensures that cluster-heads are at least of distance r from each other, which will be used in Lemma 2.1 below. To balance the energy consumption and maximize the network lifetime, re-clustering as well as gateway node rotation can be performed at predetermined times or whenever necessary. Nevertheless, it is assumed that the configurations remain static over a long time period, and the clustering overhead is amortized over multiple computation tasks.

C. Notations and Auxiliary Graphs

Denote the set of neighbors of node i in G by $\mathcal{N}(i)$, and the degree of i by $d_i = |\mathcal{N}(i)|$. The Laplacian matrix of graph G is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \text{diag}\{d_1, d_2, \dots, d_n\}$, and \mathbf{A} is the adjacency matrix of G . Let $\lambda_1(\mathbf{L}) \geq \lambda_2(\mathbf{L}) \geq \dots \geq \lambda_n(\mathbf{L}) = 0$ be the eigenvalues of \mathbf{L} in nonincreasing order. The Gerschgorin circle theorem [5] guarantees that $\lambda_1(\mathbf{L}) \leq 2d_{\max}$, with $d_{\max} = \max_{i \in V} d_i$.

Let K be the number of clusters formed by the distributed clustering algorithm above. Define the induced graph by $H = (V_H, E_H)$, where V_H consists of the set of clusters, and $(kl) \in E_H$ if and only if cluster k and l are adjacent. Let C_k denote the set of nodes in cluster k , $n_k = |C_k|$, and $c(i)$ denote the index of the cluster that node i belongs to. Let $\tilde{\mathcal{N}}_k$ denote the set of neighboring clusters of cluster k , and $D_k = |\tilde{\mathcal{N}}_k|$. The following lemma will be used in performance analysis of cluster-based algorithms.

Lemma 2.1: Using Distributed Clustering algorithm on $G(n, r)$ with $r(n) = \Omega\left(\sqrt{\frac{\log n}{n}}\right)$, we have $1 \leq D_k \leq 48$ for $1 \leq k \leq K$, and $\pi^{-1}r^{-2} \leq K \leq 2r^{-2}$.

Proof: The lower bound $D_k \geq 1$ follows from the connectivity of the network. Since the cluster-heads are at least at a distance r from each other, the circles with the cluster-heads as the centers and radius $0.5r$ are non-overlapping. Note also that, a pair of neighboring clusters must be joined by at least one link, hence the cluster-heads of neighboring clusters must lie within

distance $3r$ from each other. Within the neighborhood of radius $3.5r$ of a cluster-head, there are no more than $\left(\frac{3.5}{0.5}\right)^2$ non-overlapping circles of radius $0.5r$, which means $D_k \leq 48$. Now, consider the tessellation of the unit square into squares of side $\frac{r}{\sqrt{2}}$. Thus, every such square contains at most one cluster-head, so there are at most $2r^{-2}$ clusters. On the other hand, in order to cover the whole unit square, there must be at least $\pi^{-1}r^{-2}$ clusters. ■

Let \hat{G}_{kl} denote the clique (complete group) formed by the nodes in cluster k and l . Define graph $\hat{G} = (V, \hat{E}) = \bigcup_{(kl) \in E_H} \hat{G}_{kl}$. It is easy to see that G is a subgraph of \hat{G} . Moreover, since nodes in the same cluster and adjacent clusters all become neighbors, we shall expect that the new graph \hat{G} is much better-connected than G . We denote $\hat{\mathcal{N}}(i)$ and \hat{d}_i as the neighbor set and degree of node i in \hat{G} , $\hat{d}_{\max} = \max_{i \in V} \hat{d}_i$, and $\hat{\mathbf{L}}$ the Laplacian of \hat{G} .

III. CLUSTER-BASED CONSENSUS ALGORITHM

The initialization of both cluster-based averaging algorithms is as follows. Each node transmits its value to the corresponding cluster-head. The cluster-head computes the average value within the cluster and broadcasts it to all members. Let $\mathbf{y}(t) = [y_1(t), \dots, y_K(t)]^T$ be the average values each cluster agrees on at time t . We have

$$y_k(0) = \frac{1}{n_k} \sum_{c(i)=k} x_i(0), \quad k = 1, \dots, K, \quad (1)$$

and node i obtains after initiation $\hat{x}_i(0) = y_{c(i)}(0)$. Through information exchange of gateway nodes, each cluster-head obtains the sizes of neighboring clusters.

A. Fixed Linear Iteration

For fixed linear iteration, all nodes in the network share a global schedule. At time instant t , $t = 0, 1, \dots$, each gateway node exchanges its value with neighboring gateway nodes. It then forwards the received values to the cluster-head. The cluster-head of cluster k updates the common value in the cluster with

$$y_k(t+1) = y_k(t) + \alpha \sum_{l \in \tilde{\mathcal{N}}_k} n_l (y_l(t) - y_k(t)), \quad (2)$$

where $0 < \alpha < 1/\hat{d}_{\max}$. This value is then broadcasted to all nodes in the cluster. Node i updates its current value with $\hat{x}_i(t) = y_{c(i)}(t)$.

Definition 1. For deterministic algorithms, the ϵ -averaging time is defined as

$$T_{\text{ave}}(\epsilon) = \sup_{\mathbf{x}(0)} \inf \{t : \|\mathbf{x}(t) - x_{\text{ave}}\mathbf{1}\|_2 \leq \epsilon \|\mathbf{x}(0) - x_{\text{ave}}\mathbf{1}\|_2\} \quad (3)$$

where $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$ is the l_2 norm.

Theorem 3.1: When $0 < \alpha < \frac{2}{\lambda_1(\hat{\mathbf{L}})}$, the ϵ -averaging time of the cluster-based fixed linear iteration algorithm satisfies

$$T_{\text{ave}}(\epsilon) \leq \frac{\log \epsilon^{-1}}{\log \eta^{-1}}, \quad (4)$$

where

$$\eta \triangleq \sup_{\hat{\mathbf{x}}(t) \neq x_{\text{ave}}\mathbf{1}} \frac{\|\hat{\mathbf{x}}(t+1) - x_{\text{ave}}\mathbf{1}\|_2}{\|\hat{\mathbf{x}}(t) - x_{\text{ave}}\mathbf{1}\|_2} = \max\{\alpha\lambda_1(\hat{\mathbf{L}}) - 1, 1 - \alpha\lambda_{n-1}(\hat{\mathbf{L}})\}, \quad (5)$$

is the convergence rate. In particular, the choice of $\alpha^* = \frac{2}{\lambda_1(\hat{\mathbf{L}}) + \lambda_{n-1}(\hat{\mathbf{L}})}$ gives the optimal convergence rate $\eta^* = \frac{\lambda_1(\hat{\mathbf{L}}) - \lambda_{n-1}(\hat{\mathbf{L}})}{\lambda_1(\hat{\mathbf{L}}) + \lambda_{n-1}(\hat{\mathbf{L}})}$.

Proof: From node i 's point of view, we have at time t ,

$$\begin{aligned} \hat{x}_i(t+1) &= \hat{x}_i(t) + \alpha \sum_{j \in \hat{\mathcal{N}}(i), c(j) \neq c(i)} (\hat{x}_j(t) - \hat{x}_i(t)) \\ &= \hat{x}_i(t) + \alpha \sum_{j \in \hat{\mathcal{N}}(i)} (\hat{x}_j(t) - \hat{x}_i(t)). \end{aligned} \quad (6)$$

In matrix form we have

$$\hat{\mathbf{x}}(t+1) = \mathbf{W}\hat{\mathbf{x}}(t) \triangleq (\mathbf{I} - \alpha\hat{\mathbf{L}})\hat{\mathbf{x}}(t). \quad (7)$$

For a symmetric stochastic \mathbf{W} , the convergence rate is given by its second largest eigenvalue, i.e., $\eta = \lambda_2(\mathbf{W})$. Given $\mathbf{W} = \mathbf{I} - \alpha\hat{\mathbf{L}}$, the theorem follows readily. \blacksquare

Remark: Since $\lambda_1(\hat{\mathbf{L}}) \leq 2\hat{d}_{\max}$, a sufficient condition for convergence is $0 < \alpha < 1/\hat{d}_{\max}$.

In the following, we compare the performance of the above algorithm with the standard linear iteration with constant edge weights on the original graph [1], i.e., the weight matrix is given by $\mathbf{I} - \alpha\mathbf{L}$, with $0 < \alpha < 1/d_{\max}$. Let $b = \frac{\lambda_{n-1}(\mathbf{L})}{\lambda_1(\mathbf{L})}$ and $\hat{b} = \frac{\lambda_{n-1}(\hat{\mathbf{L}})}{\lambda_1(\hat{\mathbf{L}})}$, then the convergence rate

of the standard and cluster-based fixed linear iteration with optimal constant edge weights are respectively given by $\frac{1-b}{1+b}$ and $\frac{1-\hat{b}}{1+\hat{b}}$. For b small, we have $\log(1/\eta^*) = \log\left(\left(1 - \frac{2b}{1+b}\right)^{-1}\right) \approx 2b$. It is observed that on $G(n, r)$, due to clustering, both the smallest and the largest non-zero eigenvalue of the graph Laplacian increase (the smallest non-zero eigenvalue is known as the *algebraic connectivity*), but the ratio $\hat{b} > b$. Hence clustering accelerates convergence of the consensus algorithm (c.f. (4)). For example, on $G(1000, r(1000))$, where $r(n) = \sqrt{\frac{2\log n}{n}}$, clustering causes b to increase from 0.01 to 0.06 on average. A similar roughly 6-fold increase can be observed for other values of n . This implies that the cluster-based algorithm converges at least 6 times faster due to improved graph connectivity.

Apart from time complexity, clustering brings further gain in terms of message complexity. For distributed consensus based on the standard linear iteration algorithm, the most efficient way is to have each node broadcast its value in each iteration, hence the number of messages transmitted in each iteration is $\Theta(n)$. For the cluster-based variant, the number of messages transmitted in each iteration is on the same order as the number of gateway nodes. From Lemma 2.1, the number of neighbors of each cluster is bounded by a constant, which implies that the number of gateway nodes in the network is on the order of the number of clusters, i.e., $\Theta(r^{-2})$. Therefore, we obtain the following theorem.

Theorem 3.2: The cluster-based fixed linear iteration algorithm achieves a gain of $\Omega(nr^2)$ in message complexity to achieve ϵ -accuracy compared with the standard fixed linear iteration algorithm.

Thus, for the typical value of $r = \Theta\left(\sqrt{\frac{\log n}{n}}\right)$, an $\Omega(\log n)$ gain in message complexity can be achieved through clustering.

B. Randomized Gossip

The gossip algorithms [2], [6] are designed under the practical constraint that each node communicates with no more than one neighbor in each time slot for interference avoidance. In randomized gossip algorithms [2], neighbors are chosen at random, thus they suit the case where

nodes lack unique identity and enjoy great simplicity. Compared to fixed iterations, gossip-based algorithms also allow asynchronous operation, and achieve high stability in dynamic networks. In this section, we describe a cluster-based gossip variant, which improves the time and message complexity.

The cluster-based gossip works as follows. Assume that the cluster-head of cluster k ($1 \leq k \leq K$) has a clock ticking at a rate D_k Poisson process, and different clocks are independent. Suppose the t -th tick belongs to the cluster-head of cluster k . It randomly picks one of its neighboring cluster l with probability $1/D_k$, and communicates with it through the active link between them. The cluster-head of k and l update their values with

$$y_k(t+1) = y_l(t+1) = \frac{n_k y_k(t) + n_l y_l(t)}{n_k + n_l}. \quad (8)$$

The new value is then broadcasted, and nodes in cluster k and l update their values with $x_i(t) = y_{c(i)}(t)$. The number of message transmissions per round is at most 6 (which may be smaller if the gateway node is also the cluster-head).

Definition 2. For randomized algorithms, the ϵ -averaging time is defined as the earliest time (number of clock ticks) such that the estimation is ϵ -accurate with probability $1 - \epsilon$:

$$T_{\text{ave}}(\epsilon) = \sup_{\mathbf{x}(0)} \inf \left\{ t : \Pr \left(\frac{\|\mathbf{x}(t) - x_{\text{ave}} \mathbf{1}\|_2}{\|\mathbf{x}(0)\|_2} \geq \epsilon \right) \leq \epsilon \right\}. \quad (9)$$

Theorem 3.3: The ϵ -averaging time of the cluster-based gossip algorithm satisfies

$$T_{\text{ave}}(\epsilon) \leq \frac{6d_{\max} |E_H| \log \epsilon^{-1}}{\lambda_{n-1}(\hat{\mathbf{L}})}. \quad (10)$$

Proof: See Appendix I. ■

In the following, we demonstrate the performance gain of the cluster-based gossip algorithm over the standard gossip algorithm. To facilitate analysis and comparison, we consider a random gossip algorithm similar to the natural averaging algorithm in [2], but each edge is assumed to

be activated uniformly⁵. Thus, using the notations in [2], the expectation of the update matrix is

$$\mathbf{W} = \frac{1}{|E|} \sum_{(ij) \in E} W_{ij} = \mathbf{I} - \frac{1}{2|E|} \mathbf{L}. \quad (11)$$

By Theorem 3 in [2], the ϵ -averaging time is lower bounded by

$$T_{ave}(\epsilon) \geq \frac{0.5 \log \epsilon^{-1}}{\log \lambda_2(\mathbf{W})^{-1}} \approx \frac{|E| \log \epsilon^{-1}}{\lambda_{n-1}(\mathbf{L})}. \quad (12)$$

Comparing the upper bound in (10) with the lower bound in (12) reveals performance gain in time complexity due to clustering. Note that when $r \geq \sqrt{\frac{2 \log n}{n}}$, we have $|E_H| \leq K D_{\max} = O(r^{-2})$, and $d_{\max} = \Theta(nr^2)$, thus $d_{\max}|E_H| = \Theta(n)$. On the other hand, we have $|E| = \Theta(n^2 r^2)$. This accounts for a reduction in the numerator by a factor of $\Theta(nr^2)$. This is mainly because the value of each node is updated more often under the cluster-based gossip algorithm. The second reason lies in the improved connectivity of \hat{G} relative to G (the denominator). As an example, on $G(100, r(100))$ with $r(n) = \sqrt{\frac{2 \log n}{n}}$, the algebraic connectivity increases from 1.98 for G to 28.08 for \hat{G} on average. Fig. 1 shows the relative error $\frac{\|\mathbf{x}(t) - x_{ave} \mathbf{1}\|_2}{\|\mathbf{x}(0)\|_2}$ as a function of the number of iterations for the standard gossip algorithm and the cluster-based gossip algorithm. It is evident that clustering significantly speeds up the convergence.

Finally, the ϵ -message complexity of asynchronous gossip algorithms is of the same order as the ϵ -averaging time, since the number of transmissions per time instant is bounded by a constant independent of n . Therefore, like the cluster-based fixed linear iteration, our cluster-based gossip algorithm provides a gain of $\Omega(nr^2)$ in message complexity over standard gossip algorithms.

IV. CONCLUSION

In this paper, clustering techniques have been incorporated into distributed consensus to improve the performance. A simple non-iterative distributed clustering algorithm is provided with which nodes are clustered in a distributed manner without the requirement of global knowledge.

⁵The performances of these two algorithms are almost identical as seen in Fig. 1; they are asymptotically the same as n grows, and of the same order as the optimal averaging algorithm on the geometric random graph.

Cluster-based distributed averaging algorithms in forms of fixed linear iteration and randomized gossip are proposed. Both are shown to achieve much faster convergence and significantly reduced communication and computational complexity compared with their non-cluster-based counterparts.

APPENDIX I

PROOF OF THEOREM 3.3

Let $v_i = x_i - x_{ave}$. Define the potential function of node values as

$$\phi = \|\mathbf{x} - x_{ave}\mathbf{1}\|_2^2 = \sum_{i \in V} x_i^2 - nx_{ave}^2 = \mathbf{v}^T \mathbf{v}.$$

The reduction in the potential if cluster k and l are activated is given by

$$\Delta\phi_{kl} = \sum_{i \in C_k \cup C_l} x_i^2 - \frac{(\sum_{i \in C_k \cup C_l} x_i)^2}{n_k + n_l} = \frac{1}{n_k + n_l} \sum_{i, j \in C_k \cup C_l} (x_i - x_j)^2 = \frac{1}{n_k + n_l} \sum_{i, j \in C_k \cup C_l} (v_i - v_j)^2.$$

Thus, the expected reduction in the potential at each clock tick is

$$\mathbb{E}[\Delta\phi] = \frac{1}{|E_H|} \sum_{(kl) \in E_H} \frac{1}{n_k + n_l} \sum_{i, j \in C_k \cup C_l} (v_i - v_j)^2 \geq \frac{1}{2d_{\max}|E_H|} \sum_{(ij) \in E_{\hat{G}}} (v_i - v_j)^2 = \frac{\mathbf{v}^T \hat{\mathbf{L}} \mathbf{v}}{2d_{\max}|E_H|}.$$

The relative decrement is lower bounded by

$$\frac{\mathbb{E}[\Delta\phi]}{\phi} \geq \frac{1}{2d_{\max}|E_H|} \frac{\mathbf{v}^T \hat{\mathbf{L}} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \geq \frac{\lambda_{n-1}(\hat{\mathbf{L}})}{2d_{\max}|E_H|}.$$

Let $a = 1 - \frac{\lambda_{n-1}(\hat{\mathbf{L}})}{2d_{\max}|E_H|}$. We have $\mathbb{E}[\phi(t)] \leq a^t \phi(0) \leq a^t \mathbf{x}(\mathbf{0})^T \mathbf{x}(\mathbf{0})$. Applying the Markov's inequality, we have that for $t \geq \frac{3 \log \epsilon^{-1}}{\log a^{-1}}$,

$$\Pr\left(\frac{\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_2}{\|\mathbf{x}(\mathbf{0})\|_2} \geq \epsilon\right) = \Pr\left(\frac{\phi(t)}{\mathbf{x}(\mathbf{0})^T \mathbf{x}(\mathbf{0})} \geq \epsilon^2\right) \leq \epsilon^{-2} \frac{\mathbb{E}[\phi(t)]}{\mathbf{x}(\mathbf{0})^T \mathbf{x}(\mathbf{0})} \leq \epsilon^{-2} a^t \leq \epsilon.$$

The desired result follows from $\log a^{-1} \approx \frac{\lambda_{n-1}(\hat{\mathbf{L}})}{2d_{\max}|E_H|}$.

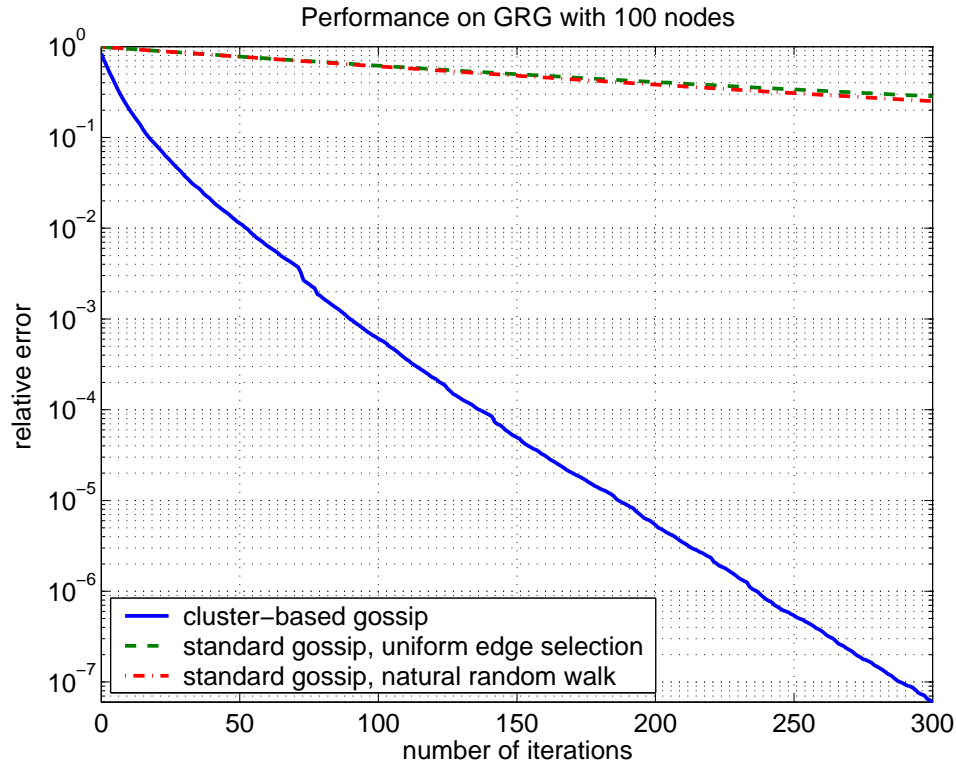


Fig. 1. Decay of l_2 relative error for standard and cluster-based gossip algorithms on $G(100, r(100))$

REFERENCES

- [1] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” in *IEEE Conf. on Decision and Control*, Maui, Hawaii, Dec. 2003.
- [2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2506–2530, 2006.
- [3] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [4] M. R. Garey and D. S. Johnson, *A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
- [5] H. E. Bell, “Gerschgorin’s theorem and the zeros of polynomials,” *Amer. Math. Monthly*, vol. 72, pp. 292–295, 1965.
- [6] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information,” in *IEEE Symp. on Foundations of Computer Science (FOCS)*, Cambridge, MA, Oct. 2003.