LADA Algorithms: Performance Lower Bound and Cluster-based Variant

Wenjun Li and Huaiyu Dai

Abstract— **The observation that certain nonreversible chains lifted from reversible ones mix substantially faster than the original chains [1], [2] has motivated our recent work [3], where a Location-Aided Distributed Averaging (LADA) algorithm is proposed to achieve fast distributed consensus. We continue our study in this paper to gain a better understanding of distributed consensus via chain lifting and explore possible improvements for LADA. First, lower bounds for the averaging time via lifting Markov chains in a wireless network with a common transmission range r, as modeled by a geometric random graph $G(n,r)$, are derived based on resistance, an invariant of Markov chains. It is shown that the constructed chain in LADA is optimal in the scaling law for the mixing parameter $T_{\rm fill}$, for Markov chains with an approximately uniform stationary distribution on $G(n,r)$. The ϵ-averaging time of $O(r^{-1}\log(\epsilon^{-1}))$ achieved by LADA is also close to its lower bound $\Omega(r^{-1})$. Finally, a cluster-based LADA (C-LADA) variant is proposed to further improve on the message complexity in dense networks. Essentially, the LADA algorithm is applied to the induced graph resulted from a distributed clustering algorithm. It is shown that the number of transmitted messages per iteration is reduced from $\Theta(n)$ to $\Theta(r^{-2})$, which translates into significant power savings for nodes.**

I. Introduction

The distributed consensus problem where nodes in a network try to reach consensus on their average value through iterative local information exchange has been vigorously investigated recently [4]–[8]. Distributed computation of the average over a network can be realized through linear iteration in the form $\mathbf{x}(t{+}1) = \mathbf{W}(t)\mathbf{x}(t)$ where $\mathbf{W}(t)$ is a graph conformant matrix, i.e., $W_{ij}(t) \neq 0$ only if there is a direct link from node i to j in the network. Typically, governing matrices in distributed consensus (or more general distributed computation) algorithms are chosen to be stochastic, which connects them closely to Markov chain theory. In both fixed and random gossip algorithms studied in [4], [5], mainly a symmetric, doubly stochastic weight matrix is used, hence the convergence time of such algorithms is closely related to the mixing time of a reversible random walk, which is usually slow due to its diffusive behavior.

It has been observed by Diaconis *et al.* [1] and Chen *et al.* [2] that nonreversible chains constructed on a "lifted" graph mix substantially faster than related reversible chains. A lifted chain $\tilde{\mathbf{P}}$ (with state space $\tilde{V}$) is constructed from a given chain $\mathbf{P}$ (with state space V) by creating multiple replica states $u_1, \cdots, u_{l_u}$ in $\tilde{V}$ corresponding to each state $u \in V$, and designing the transition probabilities conforming to $\mathbf{P}$ (i.e., $\tilde{P}_{u_i v_j} > 0$ only if $P_{uv} > 0$ or $u = v$) such that the stationary probabilities $\pi(u)$, $u \in V$ in $\mathbf{P}$ is retained (i.e., $\sum_i \tilde{\pi}(u_i) = \pi(u)$). $\mathbf{P}$ is called the collapsed chain of $\tilde{\mathbf{P}}$. The idea of nonreversible lifting lends itself naturally to fast distributed consensus algorithms: by allowing each node in a network to maintain multiple copies of its estimated average value, lifted chains with desired fast-mixing properties can be simulated. In a recent paper [3], we propose a Location-Aided Distributed Averaging (LADA) algorithm, which exploits the knowledge of directions of neighbors to construct fast-mixing nonreversible chains. Each node maintains four copies of its estimated average value at any time; each value is more likely to propagate in a specific direction, with which the diffusive behavior is suppressed. It is shown that in a connected wireless network of size n with a common transmission range r, LADA achieves an ϵ-averaging time of $O(r^{-1}\log(\epsilon^{-1}))$, while the optimal gossip algorithm requires $\Theta\left(r^{-2}\log n\right)^1$ time for the relative error to be bounded by $\epsilon = n^{-\alpha}$, $\alpha > 0$.

In this paper, we aim to continue our investigation in [3], and gain deeper insights into distributed consensus via chain lifting. In particular, we wish to explore how close LADA is to the optimum, and seek to further improve LADA in dense networks. We start with some analysis for general (especially nonreversible) Markov chains on a geometric random graph $G(n,r)$–the celebrated model for wireless networks. To this end, we introduce resistance, an invariant of Markov chains. For a reversible chain, it is known that its mixing time is at least on the order of its resistance [9]. We show that for nonreversible chains, the same bound holds only for another mixing parameter known as $T_{\rm fill}$ in [10]. Subsequently, we show that the resistance of a Markov chain with an approximately uniform stationary distribution[2] on $G(n,r)$ is $\Omega(r^{-1})$, which cannot be further reduced by lifting. Thus, we establish that for any chain with an approximately uniform stationary distribution on $G(n,r)$ and its lifted chain, $T_{\rm fill} = \Omega(r^{-1})$. The lower bound $T_{\rm fill} = \Theta(r^{-1})$ is precisely achieved by the nonreversible chain constructed in LADA. Moreover, it is shown that the ϵ-mixing time of any chain with an approximately uniform stationary distribution is $\Omega(r^{-1})$ for $\epsilon = O(\frac{1}{n})$. This is also

[1]We use the following order notations in this paper: Let $f(n)$ 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 $n \geq k$; $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ as well as $f(n) = \Omega(g(n))$.

[2]For reasons described at the beginning of Section III.B, we mainly focus on Markov chains with an approximately uniform stationary distribution in this paper.

Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695. (email: wli5, huaiyu_dai@ncsu.edu). This research was supported in part by the National Science Foundation under Grant CCF-0515164.

a lower bound for the ϵ-averaging time of the corresponding distributed consensus algorithm.

While the time complexity is certainly important for distributed algorithms, the message complexity directly determines the power consumption of the nodes. As a second contribution of this paper, we propose a cluster-based LADA (C-LADA) variant to further improve on the message complexity. This is motivated by the common assumption that nodes in some networks, such as wireless sensor networks, are densely deployed, where it is often more efficient to have co-located nodes clustered, effectively behaving as a single entity. Clustering is performed through a distributed clustering algorithm. After the initiation, the LADA algorithm is applied on the induced graph. Only inter-cluster communication and intra-cluster broadcast are needed to update the values of all nodes. The same time complexity as LADA is achieved, but the number of messages per iteration is reduced from $\Theta(n)$ to $\Theta(r^{-2})$.

Our paper is organized as follows. In Section II, we formulate the problem and discuss several important related works, including a brief introduction of LADA. In Section III, we establish some results for non-reversible Markov chains and lower bounds for the averaging time through lifting Markov chains. In Section IV, we present the C-LADA algorithm, and analyze its performance. Finally, conclusions are given in Section V.

II. Problem Formulation and Related Works

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 vector $\mathbf{x}(0) = [x_1(0), \cdots, x_n(0)]^T$ contain the initial values observed by the nodes, and $x_{\mathrm{ave}} = \frac{1}{n}\sum_{i=1}^{n} x_i$ denote the average. The goal is to compute x_{ave} in a distributed and robust fashion. Let $\mathbf{x}(t)$ be the vector containing node values at the tth iteration. Without loss of generality, we consider the set of initial values $\mathbf{x}(0) \in {\mathbb{R}^+}^n$, and define the ϵ-averaging time as

$$T_{\mathrm{ave}}(\epsilon) = \sup_{\mathbf{x}(0)\in{\mathbb{R}^+}^n} \inf\left\{t : \|\mathbf{x}(t) - x_{\mathrm{ave}}\mathbf{1}\|_1 \le \epsilon\|\mathbf{x}(0)\|_1\right\}^3 \tag{1}$$

where $\|\mathbf{x}\|_1 = \sum_i |x_i|$ is the l_1 norm[4].

We will mainly use the geometric random graph [11] to model a wireless network in our analysis. In a geometric random graph $G(n, r(n))$, n nodes are uniformly and independently distributed on a unit square $[0, 1]^2$, and $r(n)$ is the common transmission range of all nodes. It is known that the choice of $r(n) \ge \sqrt{\frac{2\log n}{n}}$ is required to ensure the graph is connected with high probability [11].

[3]For the more general case $\mathbf{x}(0) \in \mathbb{R}^n$, the corresponding expression in (1) is $\|\mathbf{x}(t) - x_{\mathrm{ave}}\mathbf{1}\|_1 \le \epsilon\|\mathbf{x}(0) - \min_i x_i(0)\mathbf{1}\|_1$.

[4]In the literature of distributed consensus, the l_2 norm has also been used in measuring the averaging time [4], [5]. We adopt the definition in (1) for ease of analysis and exposition. The same definition has been used in [8].

B. Related Works

In this section, we review several relevant works reflecting recent development on distributed consensus. Xiao and Boyd [4] derived necessary and sufficient conditions for the weight matrix $\mathbf{W}$ such that the linear iteration $\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t)$ asymptotically computes $x_{\mathrm{ave}}\mathbf{1}$ as $t \to \infty$. They formulated the fastest linear averaging problem as a semi-definite program, which is convex when $\mathbf{W}$ is restricted to be symmetric. The randomized gossip algorithm studied by Boyd *et al.* [5] realizes distributed consensus through asynchronous pairwise averaging. The authors show that the absolute $n^{-\alpha}$-averaging time ($\alpha > 0$) of the gossip algorithm is related to the mixing time of a reversible chain $\mathbf{P}$ as $\Theta(\log n + T_{\mathrm{mix}}(\mathbf{P}, \epsilon))$. As another alternative for distributed consensus, consensus propagation, a special form of Gaussian belief propagation has been proposed by Moalleimi and Roy [6]. By avoiding passing information back to where it is received, consensus propagation suppresses to some extent the diffusive nature of a reversible random walk. It requires the minimum possible time to converge on a singly connected graph, and $O(n/\epsilon \log(n/\epsilon))$ on an n-cycle. However, on well-connected graphs where node degrees are much larger than 1, the diffusive behavior is not effectively reduced with this algorithm. While the above works studied either synchronous or asynchronous parallel algorithms, the work by Savas *et al.* [7] explored distributed computation of decomposable functions through sequential algorithms, where a node does not transmit messages until it is activated by another node. Two algorithms, SIMPLE-WALK and COALESCENT are proposed, and both are shown to provide gain in message complexity at a cost of time complexity compared with gossip algorithms. Last but not the least, the independent work by Jung and Shah [8] explored the similar idea of fast distributed consensus through chain lifting as ours. Their algorithm adopts the nonreversible lifting proposed in [2], which is constructed from a multi-commodity flow of the existing chain with minimum congestion. To construct the chain each node in the network must have global knowledge of the network–in particular, the paths in the optimal multi-commodity flow that pass through itself. When a node joins or leaves the network, the entire chain needs to be re-calculated, and the state space of the new chain is of a size up to n^3.

C. LADA Algorithm

In our recent work [3], a Location-Aided Distributed Averaging (LADA) algorithm is proposed, where a nonreversible chain is formed in a distributed fashion exploiting only the direction information of neighbors, thus robust to topology change, and the size of the state space is linear in n. For completeness, we give a brief introduction of LADA in the remaining of this section. Prior to the distributed computation, each node classifies its neighbors as follows: a neighbor j of node i is said to be a Type-l neighbor of i, denoted as $j \in \mathcal{N}_i^l$, if

$$\angle(X_j - X_i) \in \left(\frac{l\pi}{2} - \frac{\pi}{4}, \frac{l\pi}{2} + \frac{\pi}{4}\right] \quad l = 0, \cdots, 3, \quad (2)$$

where $X_i = \mathrm{Re}(X_i) + i\mathrm{Im}(X_i)$ denotes the location of node i. The number of type l neighbors for node i is denoted as $d_i^l \triangleq |\mathcal{N}_i^l|$. For boundary nodes, virtual neighbors are introduced to ensure that d_i^l is roughly the same for all i and l.

The LADA algorithm works as follows. Each node i holds four pairs of values (y_i^l, w_i^l), $l = 0, \cdots, 3$, corresponding respectively to the east, north, west and south directions. The values are initialized with $y_i^l(0) = x_i(0), \quad w_i^l(0) = 1, \quad l = 0, \cdots, 3$. At time t, each node i broadcasts its four values. In turn, it updates its east value y_i^0 with

$$y_i^0(t+1) = \sum_{j \in \mathcal{N}_i^2} \frac{1}{d_j^0} \left[(1-p) y_j^0(t) + \frac{p}{2}\left(y_j^1(t) + y_j^3(t)\right)\right], \quad (3)$$

where $p = \Theta(r)$ is the probability that the associated random walk makes a turn. If i is a west boundary node, then the east value of i is a combination of values from both physical and virtual west neighbors. The north, west and south values, as well as the corresponding w values are updated in the same fashion. Node i computes its estimate of x_{ave} with $x_i(t+1) = \frac{1}{4}\sum_{l=0}^{3}(y_i^l(t+1)/w_i^l(t+1))$.

Denote $\hat{\mathbf{y}} = [\mathbf{y}_0^T, \mathbf{y}_1^T \mathbf{y}_2^T, \mathbf{y}_3^T]^T$, with $\mathbf{y}_l = [y_1^l, y_2^l, \cdots, y_n^l]^T$, and similarly denote $\hat{\mathbf{w}}$. The above iteration can be written as $\hat{\mathbf{y}}(t+1) = \mathbf{P}_1^T \hat{\mathbf{y}}(t)$ and $\hat{\mathbf{w}}(t+1) = \mathbf{P}_1^T \hat{\mathbf{w}}(t)$. It can be shown that $\mathbf{P}_1$ is an irreducible and aperiodic stochastic matrix, thus the corresponding Markov chain has a unique stationary distribution $\boldsymbol{\pi}$, which is not uniform since the incoming probabilities of a state does not sum to 1. Note that $\lim_{t\to\infty} \hat{\mathbf{y}}(t) = 4n x_{\mathrm{ave}} \boldsymbol{\pi}$ and $\lim_{t\to\infty} \hat{\mathbf{w}}(t) = 4n\boldsymbol{\pi}$. Therefore, with the aid of the auxiliary variable w, the LADA algorithm converges to the true average of node values on a finite connected 2-d network. The nonreversible random walk $\mathbf{P}_1$ is more likely to keep its direction than making a turn, and mixes much faster than reversible ones. As a result, the LADA algorithm achieves an ϵ-averaging time $T_{\mathrm{ave}}(\epsilon) = O(r^{-1}\log(\epsilon^{-1}))$ on a geometric random graph $G(n,r)$ with $r = \Omega\left(\sqrt{\frac{\log n}{n}}\right)$. Note that the averaging-time is on the same order as the mixing time of $\mathbf{P}_1$, partly due to the fact that the stationary distribution is approximately uniform through the introduction of virtual neighbors, so that all the weights w's are lower bounded by a positive constant at time $\Theta(r^{-1})$.

III. Performance of Distributed Consensus in Wireless Networks through Chain Lifting

In this section, we first introduce some parameters and analysis characterizing the mixing behavior of a general (especially non-reversible) Markov chain. Then, by deriving a lower bound for the resistance on $G(n,r)$, we show that the ϵ-averaging time of distributed consensus via lifting a Markov chain on $G(n,r)$ is at least $\Omega(r^{-1})$ when $\epsilon = O\left(\frac{1}{n}\right)$.

A. Mixing Time and Resistance

The mixing time of a Markov chain characterizes the convergence rate towards stationarity, and can be defined in various ways. The most popular definition uses the total variational distance, or equivalently, the l_1 norm. For $\epsilon > 0$, the ϵ-mixing time of an irreducible and aperiodic Markov chain $\mathbf{P}$ with stationary distribution $\boldsymbol{\pi}$ is defined as [12]

$$T_{\mathrm{mix}}(\mathbf{P}, \epsilon) = \sup_i \inf \left\{ t : \frac{1}{2}\|\mathbf{P}^t(i,\cdot) - \boldsymbol{\pi}\|_1 \le \epsilon \right\}, \quad (4)$$

where $\mathbf{P}^t(i,\cdot)$ is the t-step transition probabilities given that the start state is i. Various tools, including eigenvalues [13], coupling [14], stopping times [2], [10], [15], [16], conductance [17], canonical paths [13] and multi-commodity flow [9] have been successfully used to estimate the mixing time of reversible chains. It turns out that the analysis is much more difficult when reversibility does not hold. In this paper, we use the multi-commodity flow approach and the resistance measure [9], [18] to estimate mixing times of Markov chains without the reversibility assumption. A flow [5] in the underlying graph $G(\mathbf{P})$ of $\mathbf{P}$ is a function $f : \Gamma \to \mathbb{R}^+$ which satisfies

$$\sum_{\gamma \in \Gamma_{uv}} f(\gamma) = \pi(u)\pi(v) \quad \forall u, v \in V, u \neq v \quad (5)$$

where Γ_{uv} is the set of all simple directed paths from u to v in $G(\mathbf{P})$ and $\Gamma = \bigcup_{u\neq v} \Gamma_{uv}$. The resistance of $\mathbf{P}$, an invariant of Markov chains, is defined as the minimum congestion of all flows

$$R(\mathbf{P}) \triangleq \inf_f R(f) \triangleq \inf_f \max_e \frac{1}{Q(e)} \sum_{\gamma \in \Gamma; \gamma \ni e} f(\gamma), \quad (6)$$

where $Q(e) = Q_{ij} = \pi_i P_{ij}$ is the ergodic flow on edge $e = ij$.

It has been shown that the resistance of an ergodic reversible Markov chain satisfies $R \le 16 T_{\mathrm{mix}}(\mathbf{P}, 1/8)$ [9]. This result does not readily apply to nonreversible chains. Instead, a similar result exists if the mixing time is replaced by another parameter known as T_{fill} [10] (also known as T_{separate} [16]). For $0 < c < 1$, $T_{\mathrm{fill}}(\mathbf{P}, c)$ is defined as

$$T_{\mathrm{fill}}(\mathbf{P}, c) = \sup_i \inf \left\{ t : \mathbf{P}^t(i,\cdot) \ge (1-c)\boldsymbol{\pi} \right\}. \quad (7)$$

Lemma 3.1: For any irreducible and aperiodic Markov chain $\mathbf{P}$, the resistance satisfies

$$R(\mathbf{P}) \le \frac{T_{\mathrm{fill}}(\mathbf{P}, c)}{1-c}. \quad (8)$$

Proof: Let $t = T_{\mathrm{fill}}(\mathbf{P}, c)$, and $\Gamma_{uv}^{(t)}$ denote the set of all paths of length exactly t from u to v in the underlying graph $G(\mathbf{P})$. For each $\gamma \in \Gamma_{uv}^{(t)}$, let $p(\gamma)$ denote the probability that the random walk starting in state u takes the path γ. For each u, v and $\gamma \in \Gamma_{uv}^{(t)}$, set

$$f(\gamma) = \frac{\pi(u)\pi(v)p(\gamma)}{P^t(u,v)} \quad (9)$$

[5] An alternative and equivalent definition of a flow as a function defined on the edges of graphs can be found in [19].

and set $f(\gamma) = 0$ for all other paths $\gamma \in \Gamma$. Note that $\sum_{\gamma\in\Gamma_{uv}^{(t)}} f(\gamma) = \pi(u)\pi(v)$, and by removing cycles on all paths, we can obtain a flow f' (consisting of simple paths) from f without increasing the throughput on any edge. The flow routed by f' through e is

$$\sum_{\gamma\in\Gamma;\gamma\ni e} f'(\gamma) \le \sum_{u,v} \sum_{\gamma\in\Gamma_{uv}^{(t)},\gamma\ni e} \frac{\pi(u)\pi(v)p(\gamma)}{P^t(u,v)}$$
$$\le \frac{1}{1-c}\sum_{u,v} \sum_{\gamma\in\Gamma_{uv}^{(t)},\gamma\ni e} \pi(u)p(\gamma), \tag{10}$$

where the second inequality follows from the definition of T_{fill}. The final sum in (10) is precisely the probability that the stationary process traverses the oriented edge e within t steps, which is at most $tQ(e)$. It then follows

$$R(f') = \max_e \frac{\sum_{\gamma\in\Gamma;\gamma\ni e} f'(\gamma)}{Q(e)} \le \frac{t}{1-c}. \tag{11}$$

■

For a Markov chain $\mathbf{P}$, define its reverse chain $\overleftarrow{\mathbf{P}}$ with $\overleftarrow{P}_{ij} = \frac{\pi_j P_{ji}}{\pi_i}$. In the following, we state two lemmas which will be used to establish lower bounds for the mixing parameters.

Lemma 3.2: For any irreducible and aperiodic Markov chain $\mathbf{P}$,

$$T_{\text{fill}}\left(\mathbf{P}, \frac{1+6\epsilon}{2}\right) \le T_{\text{mix}}(\mathbf{P},\epsilon) + T_{\text{mix}}(\overleftarrow{\mathbf{P}},\epsilon). \tag{12}$$

Proof: Let $\tau = T_{\text{mix}}(\mathbf{P},\epsilon)$ and $\overleftarrow{\tau} = T_{\text{mix}}(\overleftarrow{\mathbf{P}},\epsilon)$. For any $j \in V$, consider the set $S = \{k \in V : \frac{\overleftarrow{P}^{\overleftarrow{\tau}}(j,k)}{\pi(k)} \ge \frac{1}{2}\}$. By the definition of $\overleftarrow{\tau}$ we have $\pi(S) \ge 1 - 4\epsilon$. This in turn by the definition of τ ensures that for any i, $P^t(i,S) \ge 1-6\epsilon$ for $t \ge \tau$. Thus, for any $t \ge \tau + \overleftarrow{\tau}$,

$$\begin{aligned} P^t(i,j) &\ge \sum_{k\in S} P^{t-\overleftarrow{\tau}}(i,k)P^{\overleftarrow{\tau}}(k,j) \\ &= \pi(j)\sum_{k\in S} P^{t-\overleftarrow{\tau}}(i,k)\frac{\overleftarrow{P}^{\overleftarrow{\tau}}(j,k)}{\pi(k)} \\ &\ge \frac{\pi(j)}{2}\sum_{k\in S} P^{t-\overleftarrow{\tau}}(i,k) \\ &\ge \frac{1-6\epsilon}{2}\pi(j). \end{aligned} \tag{13}$$

That is, $P^t(i,j) \ge \left(1 - \frac{1+6\epsilon}{2}\right)\pi(j)$. ■

Lemma 3.3: For any irreducible and aperiodic Markov chain $\mathbf{P}$, if $\epsilon < \pi_{\min}/2$ where $\pi_{\min} = \min_i \pi(i)$, we have

$$T_{\text{mix}}(\mathbf{P},\epsilon) \ge T_{\text{fill}}\left(\mathbf{P}, \frac{2\epsilon}{\pi_{\min}}\right). \tag{14}$$

Proof: Let $t = T_{\text{mix}}(\mathbf{P},\epsilon)$. Then for any i and j, $|P^t(i,j) - \pi(j)| \le \|\mathbf{P}^t(i,\cdot) - \boldsymbol{\pi}\|_1 \le 2\epsilon$. If $\epsilon < \pi_{\min}/2$, we have $P^t(i,j) \ge \pi(j) - 2\epsilon \ge \left(1 - \frac{2\epsilon}{\pi_{\min}}\right)\pi(j)$. ■

B. A Performance Lower Bound

Consider linear iterations with which the sum of node values is retained, i.e., $\mathbf{x}(t+1) = \mathbf{P}^T\mathbf{x}(t)$, where $\mathbf{P}$ is a stochastic matrix. For the purpose of averaging, it is desirable that the associated Markov chain possesses a uniform stationary distribution, i.e., $\mathbf{P}$ is doubly stochastic. Note that in this case, $T_{\text{mix}}(\mathbf{P}, 2\epsilon)$ and the averaging time $T_{\text{ave}}(\epsilon)$ in (1) coincide. In the LADA algorithm [3], neither the constructed chain nor its collapsed chain has a uniform stationary distribution. This is compensated for by proper scaling through weight variables. It is shown that as long as the stationary distribution of the lifted chain $\widetilde{\mathbf{P}}$ is approximately uniform, i.e., $\widetilde{\pi}(u) = \Theta(\frac{1}{|\widetilde{V}|})$, for any $u \in \widetilde{V}$ (with which the stationary distribution of its collapsed chain must also be approximately uniform), the averaging time of distributed consensus is still on the same order as the mixing time of the chain. Therefore, we will mainly investigate the class of Markov chains $\mathbf{P}$ with an approximately uniform stationary distribution. In the remaining of this section, we obtain a lower bound for the mixing time of such chains and their lifted chains on $G(n,r)$, or equivalently, for the averaging time of corresponding distributed consensus algorithms. To this end, we introduce another invariant of a Markov chain, the conductance, which measures the chance of leaving a set after a single step, and is defined as [9]

$$\Phi(\mathbf{P}) = \min_{S\subset V, 0<\pi(S)<1} \frac{Q(S,\bar{S})}{\pi(S)\pi(\bar{S})} \tag{15}$$

where $\bar{S}$ is the complement of S in V, and $Q(A,B) = \sum_{i\in A}\sum_{j\in B} Q_{ij}$.

Lemma 3.4: For any Markov chain on a geometric random graph with $\pi(v) = \Theta\left(\frac{1}{n}\right)$, $\forall v \in V$, the conductance satisfies $\Phi(\mathbf{P}) = O(r)$, and the resistance satisfies $R(\mathbf{P}) = \Omega(r^{-1})$.

Proof: Consider dividing the square with a line parallel to one of its sides into two halves S and $\bar{S}$ with $\pi(S) \approx \pi(\bar{S}) \approx 1/2$, as illustrated in Fig. 1. A node in S must lie in the shadowed region to have a neighbor in $\bar{S}$. For any such node i, $\sum_{j\in\bar{S}} P_{ij} \le 1$. Applying the Chernoff bound [20], it can be shown that when $r = \Omega\left(\sqrt{\frac{\log n}{n}}\right)$, the number of nodes in the shadowed area is upper bounded by $2rn$ with high probability (w.h.p.). Therefore, we have

$$\Phi(\mathbf{P}) < \frac{Q(S,\bar{S})}{\pi(S)\pi(\bar{S})} \le \frac{2rn\cdot\Theta\left(\frac{1}{n}\right)\cdot 1}{0.5\cdot 0.5} = \Theta(r). \tag{16}$$

That is, $\Phi(\mathbf{P}) = O(r)$. By the the max-flow min-cut theorem [9], the resistance R is related to the conductance Φ as $R \ge \frac{1}{\Phi}$, thus we have $R(\mathbf{P}) = \Omega(r^{-1})$. ■

Combining Lemma 3.1, 3.2, 3.3 and 3.4 yields the following.

Theorem 3.1: For any Markov chain $\mathbf{P}$ with an approximately uniform stationary distribution on a geometric random graph $G(n,r)$, $c \in (0,1)$ and $\epsilon > 0$, we have $T_{\text{fill}}(\mathbf{P},c) = \Omega(r^{-1})$, and $\max\{T_{\text{mix}}(\mathbf{P},\epsilon), T_{\text{mix}}(\overleftarrow{\mathbf{P}},\epsilon)\} = \Omega(r^{-1})$. Moreover, if $\epsilon = O\left(\frac{1}{n}\right)$, then $T_{\text{mix}}(\mathbf{P},\epsilon) = \Omega(r^{-1})$.

Note that the resistance cannot be reduced by lifting [2]. Thus, the above also holds for lifted chains from $G(n,r)$ with an approximately uniform stationary distribution. In the process of proving Lemma 4.1 in [3], we showed that for the LADA algorithm, the random walk $\mathbf{P}_1$ starting at any

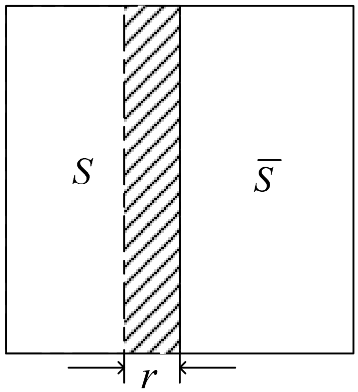

Fig. 1. Upper bound for the conductance of a Markov chain on $G(n, r)$

state visits any state i with probability at least $c_1\pi(i)$ by some time $t = \Theta(r^{-1})$, for some constant $c_1 > 0$. This implies that $T_{\text{fill}}(\mathbf{P}_1, c) = \Theta(r^{-1})$, hence LADA achieves the optimal scaling law for T_{fill}.

Now, recall a well-known result in Markov chain theory [21] that for any starting state i, and all $t = 1, 2, \cdots$, $\|\mathbf{P}^t(i,\cdot) - \pi\|_1 \leq c^{\llcorner t/T_{\text{fill}}(\mathbf{P},c)\lrcorner}$, from which it follows $T_{\text{mix}}(\mathbf{P}, \epsilon) \leq \left\{[\log(c^{-1})]^{-1}\log[(2\epsilon)^{-1}] + 1\right\} T_{\text{fill}}(\mathbf{P}, c)$. Thus, we obtained in [3] that LADA has an ϵ-averaging time of $O(r^{-1}\log\epsilon^{-1})$. For $\epsilon = \frac{1}{n^\alpha}$, $\alpha \geq 1$, $r = \Theta\left(\sqrt{\frac{\log n}{n}}\right)$, LADA improves the ϵ-mixing time of known schemes based on reversible chains by a factor of $\Omega\left(\sqrt{\frac{n}{\log n}}\right)$, but is still $O(\log n)$ away from the lower bound in Theorem 3.1. Whether the lower bound $\Omega(r^{-1})$ can be further tightened, and whether LADA achieves the optimal ϵ-averaging time in scaling law deserve further study.

IV. Cluster-based LADA

Since nodes in some wireless networks are densely deployed, it is often more efficient to have collocated nodes clustered, effectively behaving as a single entity. Only one representative node in each cluster, known as the cluster-head, is involved in the message exchange. Thus, a gain in the message complexity could be obtained, which translates directly into power savings for nodes. In this section, we introduce a cluster-based LADA (C-LADA) algorithm for this purpose, which can be simply described as follows. The nodes are clustered beforehand using a distributed clustering algorithm. The LADA algorithm is then applied to the induced graph, where the neighbor classification is based on the relative location of the cluster-heads of neighboring clusters.

A. Distributed Clustering

We begin by introducing a simple distributed clustering algorithm for general wireless networks. 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 (reaches 0), 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 and replies with the "cluster_join" message. At the end, clusters are formed such that every node belongs to one and only one cluster. The uniqueness of seeds within the neighborhood ensures that cluster-heads are at least of distance r from each other, which will be used to bound the message complexity later. We assume that clusters are formed in advance and the overhead is amortized over the multiple computations. detailed algorithm is given in Algorithm 1.

Algorithm 1 Distributed Clustering

```
K ⇐ 0 {K: number of clusters}
for all i ∈ V do
  t_i ⇐ s_i
end for
repeat
  for all i with t_i > 0 do
    t_i ⇐ t_i − 1
    if t_i = 0 then
      K ⇐ K + 1, C_K ⇐ {i} {C_k: nodes in cluster k}
      for all j ∈ N_i and with t_j > 0 do
        t_j ⇐ 0, C_K ⇐ C_K ⋃{j}
      end for
    end if
  end for
until ⋃_k C_k = V
```

We say two clusters are adjacent (or neighbors) if there is a direct link joining them. Assume that for each pair of neighbors, one single link joining them is activated. The end nodes of active links are called gateway nodes. The induced graph $\tilde{G}$ from clustering is a graph with the vertex set consisting of all cluster-heads and the edge set obtained by joining the cluster-heads of neighboring clusters. Note that the resultant topology is arbitrary, in contrast to the grid structure obtained by tessellation through a centralized scheme. In Fig. 2, we illustrate the induced graph as a result of applying our distributed clustering algorithm to a realization of G(300, r(300)), where $r(n) = \sqrt{\frac{2\log n}{n}}$.

B. C-LADA Algorithm

The neighboring clusters are classified based on the relative coordinates of the cluster-heads, according to the same rule as in the LADA algorithm described in Section II.C. The set of type-l physical neighbors of cluster m obtained through this rule is denoted as $\mathcal{N}_m^l$. If m is an east boundary cluster with no physical east neighbor, m is considered as a virtual east neighbor of itself, and similarly for other directions. The degree d_m^l is defined as the number of type-l

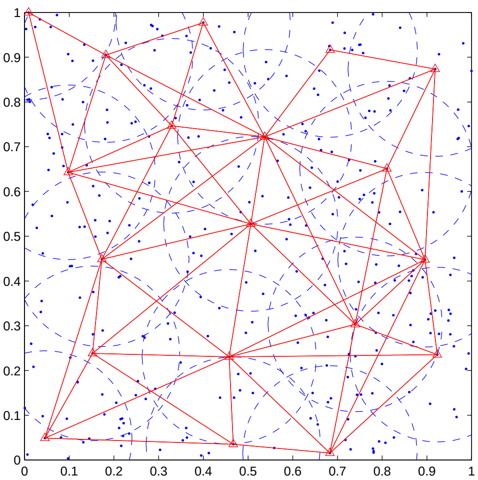

Fig. 2. Illustration of the induced graph from distributed clustering of a realization of G(300, r(300)). Nodes are indicated with small dots, cluster-heads are indicated with small triangles, cluster adjacency are indicated with solid lines, and the transmission range (not clusters) of cluster-heads are indicated with dashed circles.

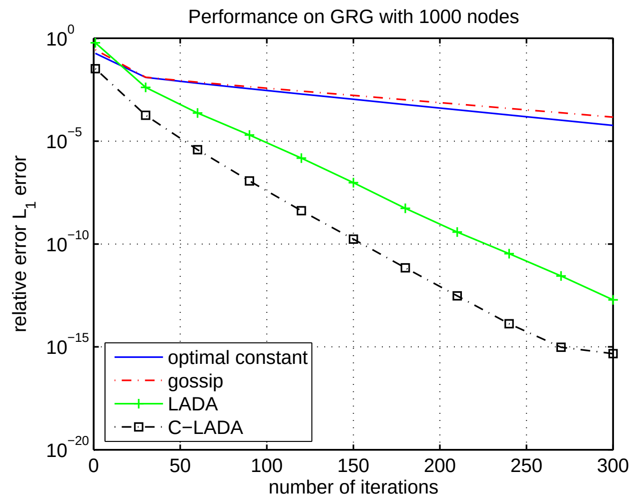

Fig. 3. Performance of distributed averaging algorithms on G(1000,r(1000))

neighbors (including virtual neighbors) for cluster m. Note that, as $n \to \infty$, the transmission circles of all cluster-heads must cover the entire network area when the distributed clustering is completed. It can be shown that for the cluster-head of any internal cluster, there must be at least one cluster-head lying within a distance $\left(2+\frac{1}{\sqrt{5}}\right)r$ in each of the four neighboring regions such that the entire network area is covered[6], and such two clusters are adjacent w.h.p. Thus, we have $d_m^l \geq 1$ for any any cluster m and direction l w.h.p.

The LADA algorithm can be readily applied to the induced graph from clustering as follows. Every cluster-head maintains four pairs of values of (y_m^l, w_m^l), $l = 0, \cdots, 3$, initialized with $y_m^l(0) = \sum_{C_i=m} x_i(0)$, and $w_m^l(0) = n_m$, $l = 0, \cdots, 3$, where C_i is the index of the cluster node i belongs to, and n_m is the number of nodes in the m-th cluster. At time t, the gateways nodes of neighboring clusters exchange values and forward the received values to the cluster-head. The cluster-head of cluster m update its east value with

$$\begin{aligned} & y_m^0(t+1) \\ = & \sum_{j \in \mathcal{N}_m^2} \frac{1}{d_j^0}\left[(1-p)y_j^0(t) + \frac{p}{2}\left(y_j^1(t) + y_j^3(t)\right)\right], \end{aligned} \quad (17)$$

where $p = \Theta(r)$. If m is a west boundary cluster having no physical west neighbor, then the sum in (17) is replaced with $(1-p)y_m^2(t) + \frac{p}{2}\left(y_m^1(t) + y_m^3(t)\right)$, i.e. its own west (rather than east), north and south values are used. In other words, the west values are bounced back and become east values when they reach the west boundary, so that the information continues to propagate. The north, west, south values as well as the w values are updated in a similar manner. The y and w values are then broadcast to the cluster members. Every node computes the estimate of the average with $x_i(t) = \frac{1}{4}\sum_{l=0}^{3}(y_{C_i}^l / w_{C_i}^l)$.

Using our argument on LADA, it is easy to show that $\mathbf{x}(t)$ converges to $x_{\text{ave}}\mathbf{1}$ as $t \to \infty$. The performance of C-LADA on the geometric random graph can be analyzed following a similar argument as for LADA, and it can be shown that C-LADA also achieves an ϵ-averaging time of $O(r^{-1}\log(\epsilon^{-1}))$. We simulated the performance of the C-LADA and LADA, along with fixed iteration with optimal constant edge weights [4], and randomized gossip [5] where a node chooses one of its neighbors with equal probability (for a fair comparison, the absolute averaging time of the asynchronous gossip is used). Fig. 3 illustrates the relative l_1 error decay on a realization of $G(1000, r(1000))$, where $r(n) = \sqrt{\frac{2\log n}{n}}$. It can be seen that both LADA and C-LADA significantly outperform algorithms based on reversible chains. Since nodes in neighboring clusters also become neighbors, the network connectivity is improved through clustering, and C-LADA also performs slightly better than LADA.

C. Message Complexity

Finally, we demonstrate that C-LADA considerably reduces the message complexity, and hence the energy consumption. For LADA, each node must broadcast its values during each iteration, hence the number of messages transmitted in each iteration is $\Theta(n)$. For C-LADA, there are three types of messages: transmissions between gateway nodes, transmissions from the gateway nodes to cluster-heads and broadcasts by the cluster-heads. Thus, the number of messages transmitted in each iteration is on the same order

[6] This is obtained as follows: consider the east neighboring region of cluster m, and the extreme case where there are two neighboring clusters with respective cluster-heads lying on the 45 degree borderline of the north and the east region, and that of the south and the east region (they are counted as north and south neighbors respectively). The distance of both clusters from the cluster m can be varied, causing their collective covered area to vary. It can be shown that any point beyond $\left(1+\frac{1}{\sqrt{5}}\right)r$ to the east of the cluster-head of m cannot be covered by these two clusters. Thus there must be at least one other cluster-head lying with a distance $\left(2+\frac{1}{\sqrt{5}}\right)r$ from the cluster-head of m that covers it.

as the number of gateway nodes, which is between $Kd_{\min}$ and $Kd_{\max}$, where K is the number of clusters, and $d_{\min}$ and $d_{\max}$ are respectively the minimum and the maximum number of neighboring clusters in the network.

Lemma 4.1: Using the Distributed Clustering algorithm in Section IV.A, the number of neighboring clusters for any cluster m satisfies $4 \leq d_m \leq 48$, and the number of clusters satisfies $\pi^{-1}r^{-2} \leq K \leq 2r^{-2}$.

Proof: The lower bound $d_m \geq 4$ follows from $d_m^l \geq 1$ for any m and l. Note that the cluster-heads are at least at a distance r from each other. Hence, the circles with the cluster-heads as the centers and radius $0.5r$ are non-overlapping. Note also that, for a cluster m, the cluster-heads of all its neighboring clusters must lie within distance $3r$ from the cluster-head of m. 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 $0.5r$. This means that the number of neighboring clusters is upper bounded by 48.

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. ■

The theorem below on the message complexity follows immediately.

Theorem 4.1: The ϵ-message complexity, defined as the total number of messages transmitted in the network to achieve ϵ-accuracy, is $O(nr^{-1}\log(\epsilon^{-1}))$ for the LADA algorithm, and $O(r^{-3}\log(\epsilon^{-1}))$ for the C-LADA algorithm.

It is also of interest to compare our proposed algorithms with the sequential algorithms proposed in [7]. It has been shown that the message complexity of the algorithm SIMPLE-WALK in [7] is the same as the time complexity, which is equal to the cover time of the network graph. It is known that the cover time for geometric random graphs is $\Theta(n\log n)$ w.h.p. when $r = \Omega\left(\sqrt{\frac{\log n}{n}}\right)$ [22]. Thus, for $\epsilon = \frac{1}{n^\alpha}$, $\alpha > 0$ and $r = \Theta\left(\sqrt{\frac{\log n}{n}}\right)$, C-LADA suffers from a loss of $O\left(\sqrt{\frac{n}{\log^3 n}}\right)$ in message complexity, but gains $\Omega\left(\sqrt{n\log n}\right)$ in time complexity compared with SIMPLE-WALK.

V. Conclusion

In this paper, we continue our work in [3] to investigate distributed consensus through lifting Markov chains, and seek to further improve the LADA algorithm in dense networks. It is shown that the nonreversible chain constructed in LADA achieves the lower bound of the mixing parameter T_{fill} in the scaling law, i.e., $T_{\text{fill}} = \Theta(r^{-1})$, for Markov chains with an approximately uniform stationary distribution on $G(n,r)$. Moreover, the ϵ-averaging time of $O(r^{-1}\log(\epsilon^{-1}))$ for LADA is close to the $\Omega(r^{-1})$ lower bound for consensus algorithms based on such chains. As another contribution of this paper, we propose a cluster-based LADA (C-LADA) variant, where LADA is applied on the induced graph resulted from a distributed clustering algorithm. Our analysis reveals that C-LADA provides significant improvement on the message complexity, and numerical results demonstrate that slight gain in time complexity is also achieved due to improved network connectivity.

References

[1] P. Diaconis, S. Holmes, and R. M. Neal, "Analysis of a non-reversible markov chain sampler," Biometrics Unit, Cornell University, Tech. Rep. BU-1385-M, 1997.

[2] F. Chen, L. Lovász, and I. Pak, "Lifting markov chains to speed up mixing," in *31st Annual ACM Symposium on Theory of Computing (STOC'99)*, Atlanta, Georgia, May 1999.

[3] W. Li and H. Dai, "Accelerating distributed consensus via lifting Markov chains," in *Proc. IEEE Int. Symp. Information Theory (ISIT) 2007*, Nice, France, June 2007.

[4] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *IEEE Conf. on Decision and Control*, Maui, Hawaii, Dec. 2003.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2506–2530, 2006.

[6] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[7] O. Savas, M. Alanyali, and V. Saligrama, "Randomized sequential algorithms for data aggregation in sensor networks," in *Conference on Information Sciences and Systems (CISS)*, Princeton University, NJ, Mar. 2006.

[8] K. Jung and D. Shah, "Fast gossip via nonreversible random walk," in *IEEE Information Theory Workshop (ITW'06)*, Punta del Este, Uruguay, Mar. 2006.

[9] A. Sinclair, "Improved bounds for mixing rates of markov chains and multicommodity flow," *Combinatorics, Probability and Computing*, vol. 1, pp. 351–370, 1992.

[10] L. Lovász and P. Winkler, "Reversal of markov chains and the forget time," *Combinatorics, Probability and Computing*, vol. 7, pp. 189–204, 1998.

[11] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[12] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*, online book available at http://www.stat.berkeley.edu/users/aldous/RWG/book.html.

[13] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of markov chains," *Annals of Probability*, vol. 1, pp. 36–61, 1991.

[14] D. Aldous, "Random walks on finite groups and rapidly mixing markov chains," *Springer Lecture Notes in Mathematrics 986*, pp. 243–297, 1983.

[15] L. Lovász and P. Winkler, "Mixing times." [Online]. Available: citeseer.ist.psu.edu/53421.html

[16] D. Aldous, L. Lovász, and P. Winkler, "Mixing times for uniformly ergodic markov chains," *Stochastic Processes and Their Applications*, vol. 71, no. 2, pp. 165–185, Nov. 1997.

[17] A. Sinclair and M. Jerrum, "Approximate counting, uniform generation and rapidly mixing markov chains," *Information and Computation*, vol. 82, pp. 93–133, 1998.

[18] V. Guruswami, "Rapidly mixing markov chains: a comparison of techniques (a survey)," available at http://www.cs.washington.edu/homes/venkat/pubs/papers/markov-survey.ps.

[19] B. Bollobas, *Graph Theory: An Introductory Course*. New York: Springer-Verlag, 1979.

[20] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics*, vol. 23, pp. 493–507, 1952.

[21] R. M. Neal, "Probabilistic inference using markov chain monte carlo methods," Dept. of Computer Science, University of Toronto, Tech. Rep. CRG-TR-93-1, 1993. [Online]. Available: http://www.cs.utoronto.ca/ radford/.

[22] C. Avin and G. Ercal, "On the cover time and mixing time of random geometric graphs," to appear in Journal of Theoretical Computer Science.