

Distributed Averaging Via Lifted Markov Chains

Kyomin Jung, Devavrat Shah, and Jinwoo Shin

Abstract—Motivated by applications of distributed linear estimation, distributed control, and distributed optimization, we consider the question of designing linear iterative algorithms for computing the average of numbers in a network. Specifically, our interest is in designing such an algorithm with the fastest rate of convergence given the topological constraints of the network. As the main result of this paper, we design an algorithm with the fastest possible rate of convergence using a nonreversible Markov chain on the given network graph. We construct such a Markov chain by transforming the standard Markov chain, which is obtained using the Metropolis–Hastings method. We call this novel transformation *pseudo-lifting*. We apply our method to graphs with geometry, or graphs with doubling dimension. Specifically, the convergence time of our algorithm (equivalently, the mixing time of our Markov chain) is proportional to the diameter of the network graph and hence optimal. As a byproduct, our result provides the fastest mixing Markov chain given the network topological constraints, and should naturally find their applications in the context of distributed optimization, estimation and control.

Index Terms—Consensus, lifting, linear averaging, Markov chain, nonreversible, pseudo-lifting, random walk.

I. INTRODUCTION

THE recently emerging network paradigms such as sensor networks, peer-to-peer networks, and surveillance networks of unmanned vehicles have led to the requirement of designing distributed, iterative, and efficient algorithms for estimation, detection, optimization, and control. Such algorithms provide scalability and robustness necessary for the operation of such highly distributed and dynamic networks. In this paper, motivated by applications of linear estimation in sensor networks [17], [7], [23], [31], information exchange in peer-to-peer networks [21], [27] and reaching consensus in unmanned vehicles [16], we consider the problem of computing the average of numbers in a given network in a distributed manner. Specifically, we consider the class of algorithms for computing the average using distributed linear iterations. In applications of interest, the rate of convergence of the algorithm strongly affects its performance. For example, the rate of convergence of the algorithm determines the agility of a distributed estimator to track the desired value [7] or the error in the distributed optimization algorithm [28]. For these reasons, designing algorithms with fast rate of convergence is of a great

Manuscript received July 16, 2008; revised June 03, 2009. Current version published December 23, 2009. This work was supported in part by the National Science Foundation (NSF) projects HSD 0729361, CNS 0546590, TF 0728554 and the DARPA ITMANET project.

The authors are with Laboratory for Information and Decision Systems, the Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: kmjung@mit.edu; devavrat@mit.edu; jinwoos@mit.edu).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques. Color version of Figure 1 in this paper is available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2009.2034777

recent interest [7], [3], [11] and the question that we consider in this paper.

A network of n nodes whose communication graph is denoted by $G = (V, E)$, where $V = \{1, \dots, n\}$ and $E = \{(i, j) : i \text{ and } j \text{ can communicate}\}$. Each node has a distinct value and our interest is designing a distributed iterative algorithm for computing the average of these numbers at the nodes. A popular approach, started by Tsitsiklis [31], involves finding a nonnegative valued $n \times n$ matrix $P = [P_{ij}]$ such that

- a) P is graph conformant, i.e., if $(i, j) \notin E$ then $P_{ij} = 0$;
- b) $\mathbf{1}^T P = \mathbf{1}^T$, where $\mathbf{1} = [1]$ is the (column) vector of all components 1;
- c) $P^t \mathbf{x} \rightarrow x_{\text{ave}} \mathbf{1}$ as $t \rightarrow \infty$ for any $\mathbf{x} \in \mathbb{R}_+^n$, where $x_{\text{ave}} = (\sum_{i=1}^n x_i)/n$.

This is equivalent to finding an irreducible, aperiodic random walk on graph G with the uniform stationary distribution.

The quantity of interest, or the performance of algorithm, is the time it takes for the algorithm to get close to $x_{\text{ave}} \mathbf{1}$ starting from any \mathbf{x} . Specifically, given P , define the ε -computation time of the algorithm as

$$T_\varepsilon(P) = \inf \left\{ t : \forall \mathbf{x} \in \mathbb{R}_+^n, \frac{\|P^t \mathbf{x} - x_{\text{ave}} \mathbf{1}\|_\infty}{x_{\text{ave}}} \leq \varepsilon \right\}. \quad (1)$$

It is well known that $T_\varepsilon(P)$ is proportional¹ to the mixing time, denoted as $\mathcal{H}(P)$, of the random walk with transition matrix P . Thus, the question of interest in this paper is to find a graph conformant P with the smallest computation time or, equivalently, a random walk with the smallest mixing time. Indeed, the question of designing a random walk on a given graph with the smallest mixing time in complete generality is a well-known unresolved question.

The standard approach of finding such a P is based on the method of Metropolis [26] and Hastings [13]. This results in a *reversible* random walk P on G . The mixing time $\mathcal{H}(P)$ is known to be bounded as

$$\frac{1}{\Phi(P)} \leq \mathcal{H}(P) \leq O\left(\frac{\log n}{\Phi^2(P)}\right)$$

where $\Phi(P)$ denotes the conductance of P . Now, for *expander* graphs, the resulting P induced by the Metropolis–Hastings method is likely to have $\Phi(P) = \Theta(1)$ and hence the mixing time is $O(\log n)$ which is *essentially* the fastest possible. For example, a random walk $P = [1/n]$ on the complete graph has $\Phi(P) = 1/2$ with mixing time $O(1)$. Thus, the question of interest is reasonably resolved for graphs that are expanding.

Now the graph topologies arising in practice, such as those in wireless sensor network deployed in some geographic area [7], [11] or a nearest neighbor network of unmanned vehicle [30], do possess *geometry* and are far from being expanders. A

¹Lemma 8 states the precise relation. Known terms, such as mixing time, that are used here are defined in Section II-A.

Fast averaging algorithms

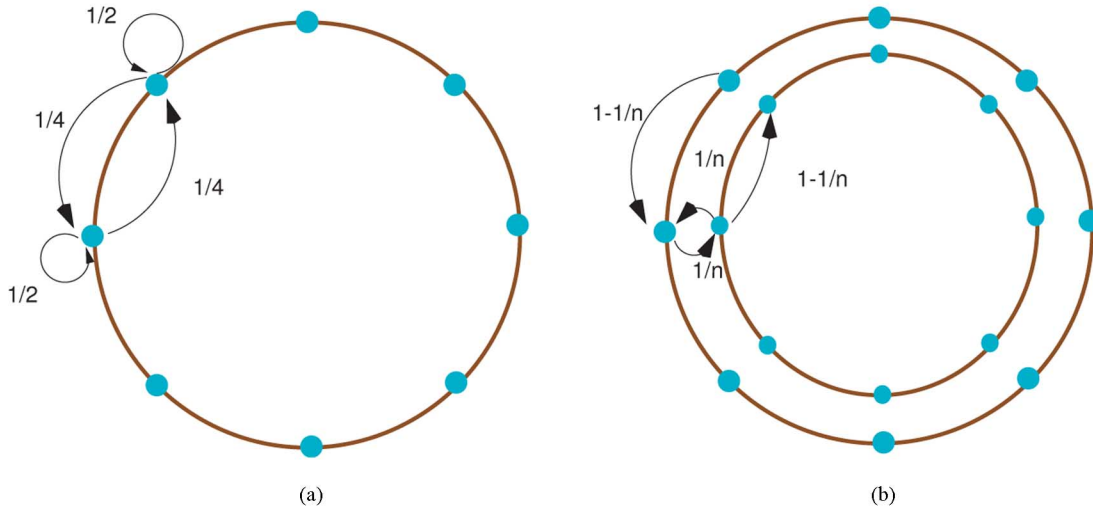


Fig. 1. (a): P_1 on the ring graph G_1 . (b): P_2 on the lifted ring graph G_2 .

simple example of graph with geometry is the *ring* graph of n nodes as shown in Fig. 1. The Metropolis–Hastings method will lead to P_1 shown in Fig. 1(a). Its mixing time is $O(n^2 \log n)$ and no smaller than $\Omega(n^2)$ (e.g., see [5]). More generally, the mixing time of any reversible random walk on the ring graph is lower-bounded by $\Omega(n^2)$ [4] for its mixing time. Note that the diameter of the ring graph is n and obviously no random walk can mix faster than the diameter. Hence, *a priori* it is not clear if the fastest mixing time is n^2 or n or something in between: that is, does the smallest mixing time of the random walk on a typical graph G scale like the diameter of G , the square of the diameter or a power of the diameter in $(1, 2)$?

In general, in most cases of interest the mixing time of the reversible walk P scales like $1/\Phi^2(P)$. The conductance $\Phi(P)$ relates to diameter D of a graph G as $1/\Phi(P) \geq D$. Therefore, in such situations the mixing time of random walk based on the Metropolis–Hastings method is likely to scale like D^2 , the square of the diameter. Indeed, Diaconis and Saloff-Coste [29] established that for a certain class of graphs with *geometry*, the mixing time of any reversible random walk scales like at least D^2 and it is achieved by the Metropolis–Hastings’ approach. Thus, reversible random walks result in rather poor performance for graphs with geometry i.e., their mixing time is far from our best hope, the diameter D .

Motivated by this, we wish to undertake the following reasonably ambitious question in this paper: *is it possible to design a random walk with mixing time of the order of diameter D for any graph?* We will answer this question in affirmative by producing a novel construction of nonreversible random walks on the *lifted* version of graph G . And thus, we will design iterative averaging algorithms with the fastest possible rate of convergence.

A. Related Work

In an earlier work, Diaconis, Holmes, and Neal [10] introduced a construction of a nonreversible random walk on the ring (and more generally ring-like) graph. This random walk runs on

the *lifted* ring graph, which is described as G_2 in Fig. 1(b). Here, by lifting we mean making additional copies of the nodes of the original graph and adding edges between some of these copies while preserving the original graph topology. Fig. 1(b) explains the construction in [10] for the ring graph. Note that each node has two copies and the lifted graph is essentially composed of two rings: an inner ring and an outer ring. The transition on the inner circle forms a clockwise circulation and the transition on the outer circle forms a counterclockwise circulation. And the probability of changing from the inner circle and *vice versa* are $1/n$ each time. By defining transitions in this way, the stationary distribution is also preserved; i.e., the sum of stationary distributions of copies is equal to the stationary distribution of their original node. Somewhat surprisingly, the authors [10] proved that this nonreversible random walk has the linear mixing time $O^*(n)$.² Thus, effectively (i.e., up to $\log n$ factor) the mixing time is of the order of the diameter n . It should be noted that because lifting preserves the graph topology and the stationary distribution, it is possible to simulate this lifted random walk on the original graph by expanding the *state* appropriately, with the desired output. Equivalently, it is possible to use a lifted random walk for linear averaging by running iterations with extra states.³

The following question arose from the work of [10]: given graph G and random walk P on G , is it possible to design a nonreversible random walk on the lifted version of G which mixes subsequently faster than P ? Can it mix in $O(D)$? This question was addressed in a subsequent work by Chen, Lovász, and Pak [8]. They provided an explicit construction of a random walk on a lifted version of G with mixing time $O^*(1/\Phi(P))$. Further, they showed that, under the notion of lifting (implicitly introduced by [10] and formalized in [8]), it is not possible to design such a lifted random walk with mixing time smaller than $\Omega(1/\Phi(P))$.

²For a function $f : \mathbb{N} \rightarrow \mathbb{R}^+$, $O^*(f(n)) := O(f(n)\text{poly}(\log n))$.

³The details are given in Section V.

Now note that $1/\Phi(P)$ can be much larger than the diameter D . As a simple example, consider a ring graph with P exactly the same as that in Fig. 1(a), but with a difference that for two edges the transition probabilities are $\delta(n)$ instead of $1/4$ (and the transition probabilities of endpoints of these edges appropriately adjusted). Then, it can be checked that $1/\Phi(P)$ is $\Omega(n/\delta(n))$ which can be arbitrarily poor compared to the diameter n by choosing small enough $\delta(n)$. A more interesting example showing this poorer scaling of $1/\Phi(P)$ compared to diameter, even for the Metropolis–Hastings’ style construction, is presented in Section III in the context of a “Barbell graph” (see Fig. 2). Thus, the lifting approach of [10], [8] cannot lead to a random walk with mixing time of the order of diameter and hence the question of existence or design of such a random walk remains unresolved.

As noted earlier, the lifted random walk can be used to design iterative algorithms (for computing an average) on the original graph since the topology of the lifted graph and the stationary distribution of the lifted random walk “projects back” onto those of the original graph and the random walk, respectively. However, running algorithm based on lifted random walks on the original graph requires additional states. Specifically, the lifted random walk based algorithm can be simulated on the original graph by running multiple *threads* on each node. Specifically, the number of operations performed per iteration across the network depends on the size⁴ of the lifted walk (or graph). In the construction of [8] for a general graph, this issue about the size of the lifted walk was totally ignored as the authors’ interest was only the time complexity, not the size. Therefore, even though time may reduce under the construction of [8] the overall cost (equal to the product of time and size) may not be reduced; or even worse, it may increase.

Therefore, from the perspective of the application of iterative algorithms we need a notion of lifting that leads to a design of a random walk that has (a) mixing time of the order of diameter of the original graph and (b) the smallest possible size.

B. Our Contributions

In this paper, we answer the above stated question affirmatively. As noted earlier, the notion of lifting of [10], [8] cannot help in answering this question. For this reason, we introduce a notion of *pseudo-lifting* which can be thought of as a relaxation of the notion of lifting. Like lifting, the notion of pseudo-lifting preserves the topological constraints of the original graph. But the relaxation comes in preserving the stationary distribution in an approximate manner. However, it should be noted that is still possible to use the pseudo-lifted random walk to perform the iterative algorithm without any approximation errors (or to sample objects from a stationary distribution without any additional errors) since the stationary distribution of pseudo-lifting under a restricted projection provides the original stationary distribution exactly. Thus, operationally our notion of pseudo-lifting is as effective as lifting.

First, we use pseudo-lifting to design a random walk with mixing time of the order of diameter of a given graph with the

desired stationary distribution. To achieve this, we first use the Metropolis–Hastings method to construct a random walk P on the given graph G with the desired stationary distribution. Then, we *pseudo-lift* this P to obtain a random walk with mixing time of the order of diameter of G . This approach is stated as Theorem 5.

As discussed earlier, the utility of such constructions lies in the context of graphs with *geometry*. The graphs with (fixed) finite *doubling dimension*, introduced in [2], [14], [12], [9], serve as an excellent model for such a class of graphs. Roughly speaking, a graph has doubling dimension ρ if the number of nodes within the shortest path distance r of any node of G is $O(r^\rho)$ (i.e., polynomial growth of the neighborhood of a node). We apply our construction of pseudo-lifting to graphs with finite doubling dimension ρ to obtain a random walk with mixing time of the order of diameter D . In order to address the concern with expansion in the size of the pseudo-lifted graph, we use the geometry of the original graph explicitly. Specifically, we reduce the size of the lifted graph by a clever combination of clustering, geometry and pseudo-lifting. This formal result is stated as follows and its proof is in Section VI-C.

Theorem 1: Consider a connected graph G with doubling dimension ρ and diameter D . It is possible to explicitly construct a pseudo-lifted random walk on G with mixing time $O(D)$ chain and size $O(Dn^{1-\frac{1}{1+\rho}})$.

As a specific example, consider a d -dimensional grid whose doubling dimension is d . The Metropolis–Hastings method has mixing time $\Omega(n^{2/d})$, compared to our construction with mixing time $O(n^{1/d})$. Further, our construction leads to an increase in size of the random walk only by $O(n^{1/d(d+1)})$ factor. That is, pseudo-lifting is optimal in terms of the number of iterations, which is equal to diameter, and in terms of cost per iteration it is lossy by a relatively small amount, for example, $O(n^{1/d(d+1)})$ for the d -dimensional grid.

In general, we can use pseudo-lifting to design iterative algorithms for computing the average of given numbers on the original graph itself. We describe a precise *implementation* of such an algorithm in Section V. The use of pseudo-lifting, primarily effective for a class of graphs with geometry, results in the following formal result whose proof is in Section V-B.

Theorem 2: Consider a given connected graph G with diameter D and each node with a distinct value. Then (using a pseudo-lifted random walk) it is possible to design an iterative algorithm whose ε -computation time is $T_\varepsilon = O^*(D \log \frac{1}{\varepsilon})$. Further, if G has doubling dimension ρ , then the network-wide total number of operations (essentially, additions) per iteration of the algorithm is $O(Dn^{1-\frac{1}{\rho+1}})$.

As a specific example, recall a d -dimensional grid with doubling dimension d and diameter $n^{1/d}$. The Metropolis–Hastings method will have mixing time $\Omega(n^{2/d})$ and per iteration number of operations $\Theta(n)$. Therefore, the number of total operations is $O(n^{1+\frac{2}{d}})$ (even the randomized gossip algorithm of [7] will have this total cost). Compared to this, Theorem 2 implies the number of iterations would be $O(n^{1/d})$ and per iteration cost would be $O(n^{1+\frac{1}{d(d+1)}})$. Therefore, the total

⁴In this paper, the size of a random walk (resp., graph) is the number of nonzero entries in its transition matrix (resp., number of edges in the graph).

cost is $O(n^{1+\frac{d+2}{d+1}})$ which is essentially close to $O(n^{1+1/d})$ for large d . Thus, it strictly improves performance over the Metropolis–Hastings method by roughly $n^{1/d}$ factor. It is worth nothing that no algorithm can have the number of total operations less than $\Omega(n^{1+1/d})$ and the number of iterations less than $\Omega(n^{1/d})$.

For the application of interest of this paper, it was necessary to introduce a new notion of lifting and indeed we found one such notion, i.e., pseudo-lifting. In general, it is likely that for certain other applications such a notion may not exist. For this reason, we undertake the question of designing a lifted (not pseudo-lifted) random walk with the smallest possible size since the size (as well as the mixing time) decides the cost of the algorithm that uses lifting. Note that the average-computing algorithm in Section V can also be implemented via lifting instead of pseudo-lifting, and the size of lifting leads to the total number of operations.⁵ As the first step, we consider the construction of Chen, Lovász, and Pak [8]. We find that it is rather *lossy* in its size. Roughly speaking, their construction tries to build a logical *complete* graph topology using the underlying graph structure. In order to construct one of n^2 edges of this complete graph topology, they use a solution of a flow optimization problem. This solution results in multiple paths between a pair of nodes. Thus, in principle, their approach can lead to a very large size. In order to reduce this size, we use two natural ideas: one, use a sparse expander graph instead of the complete graph and two, use a solution of unsplittable flows [20]. Intuitively, this approach seems reasonable but in order to make it work, we need to overcome rather nontrivial technical challenges. To address these challenges, we develop a method to analyze hybrid nonreversible random walks, which should be of interest in its own right. The formal result is stated as follows and see Section VI for its complete proof.

Theorem 3: Consider a given connected graph G with a random walk P . Then, there exists a lifted random walk with mixing time $O^*(1/\Phi(P))$ and size $O^*(|E(P)|/\Phi(P))$, where

$$E(P) = \{(i, j) : P_{ij} \neq 0 \text{ or } P_{ji} \neq 0\}.$$

Note that the lifted random walk in [8] has size $\Omega(n^2/\Phi(P))$, hence our lifting construction leads to the reduction of its size by $\Theta(n)$ factor when G is sparse.⁶ Finally, we note that the methods developed for understanding the expander-based construction (and proof of Theorem 3) can be useful in making pseudo-lifting more *robust*, as discussed in Section VII.

II. PRELIMINARIES AND BACKGROUNDS

A. Key Notions and Definitions

In this paper, $G = (V, E)$ is a given graph with n nodes, i.e., $|V| = n$. We may use $V(G)$ to represent vertices of V of G . P always denotes a transition matrix of a graph conformant random walk (or Markov chain) on G with its stationary distribution π , i.e., $P_{ij} > 0$ only if $(i, j) \in E$, and $\pi^T P = \pi^T$.

⁵One can derive its explicit performance bound as Theorem 2. It turns out that lifting is worse than pseudo-lifting in its performance, but it is more robust in its construction.

⁶A graph $G = (V, E)$ is sparse if $|E| = O(|V|)$.

We will use the notion of “Markov chain” or “random walk” depending on which notion is more relevant to the context. The reverse chain P^* of P is defined as: $P_{ij}^* = \pi_j P_{ji} / \pi_i$ for all $(i, j) \in E$. We call P *reversible* if $P = P^*$. Hence, if π is uniform⁷, P is a symmetric matrix. The conductance of P is defined as

$$\Phi(P) = \min_{S \subset V} \frac{\sum_{i \in S, j \in V \setminus S} \pi_i P_{ij}}{\pi(S)\pi(V \setminus S)}$$

where $\pi(A) = \sum_{i \in A} \pi_i$.

Although there are various (mostly equivalent) definitions of Mixing time that are considered in the literature based on different measures of the distance between distributions, we primarily consider the definition of Mixing time from the stopping rule. A stopping rule Γ is a stopping time based on the random walk of P : at any time, it decides whether to stop or not, depending on the walk seen so far and possibly additional coin flips. Suppose, the starting node w^0 is drawn from distribution σ . The distribution of the stopping node w^Γ is denoted by $\sigma^\Gamma = \tau$ and call Γ as a stopping rule from σ to τ . Let $\mathcal{H}(\sigma, \tau)$ be the infimum of mean length over all such stopping rules from σ to τ . This is well-defined as there exists the following stopping rule from σ to τ : *select i with probability τ_i and walk until getting to i* . Now, we present the definition of the (stopping rule based) Mixing time \mathcal{H} .

Definition 1 (Mixing Time): $\mathcal{H} = \max_\sigma \mathcal{H}(\sigma, \pi)$.

Therefore, to bound \mathcal{H} , we need to design a stopping rule whose distribution of stopping nodes is π .

B. Metropolis–Hastings Method

The Metropolis–Hastings method (or Glauber dynamics [19]) has been extensively studied in recent years due to its local constructibility. For a given graph $G = (V, E)$ and distribution π on V , the goal is to produce a random walk P on G whose stationary distribution is π . The underlying idea of the random walk produced by this method is choosing a neighbor j of the current vertex i at uniformly random and moving to j depending on the ratio between π_i and π_j . Hence, its explicit transition matrix P is as follows:

$$P_{ij} = \begin{cases} \frac{1}{2d} \min \left\{ \frac{\pi_j}{\pi_i}, 1 \right\}, & \text{if } (i, j) \in E \\ 0, & \text{if } (i, j) \notin E \text{ and } i \neq j \\ 1 - \sum_{k \neq i} P_{ik}, & \text{if } i = j \end{cases}$$

where d_i is a degree of vertex i and $d = \max_i d_i$. It is easy to check that $\pi^T P = \pi^T$ and P is reversible.

C. Lifting

As stated in the Introduction, motivated by a simple ring example of Diaconis *et al.* [10], Chen *et al.* [8] use the following notion of lifting.

Definition 2 (Lifting): A random walk \hat{P} on graph $\hat{G} = (\hat{V}, \hat{E})$ is called a lifting of random walk P on graph $G = (V, E)$ if there exists a many-to-one function $f : \hat{V} \rightarrow V$ such that the following holds: (a) for any $\hat{u}, \hat{v} \in \hat{V}$, $(\hat{u}, \hat{v}) \in \hat{E}$ only

⁷ π is uniform when $\pi_i = 1/n, \forall i$.

if $(f(\hat{u}), f(\hat{v})) \in E$; (b) for any $u, v \in V$, $\pi(u) = \hat{\pi}(f^{-1}(u))$, and $Q(u, v) = \hat{Q}(f^{-1}(u), f^{-1}(v))$. Here Q and \hat{Q} are ergodic flow matrices for P and \hat{P} respectively.

Here, the ergodic flow matrix $Q = [Q_{ij}]$ of P is defined as: $Q_{ij} = \pi_i P_{ij}$. It satisfies: $\sum_{i,j} Q_{ij} = 1$, $\sum_i Q_{ij} = \sum_i Q_{ji}$, and $\sum_i Q_{ij} = \pi_j$. Conversely, every nonnegative matrix Q with these properties defines a random walk with the stationary distribution π . In this paper, \hat{P} means a lifted (or pseudo-lifted) random walk of P . Similarly, \hat{G} , \hat{V} , \hat{E} and $\hat{\pi}$ are the lifted (or pseudo-lifted) versions of their original one.

Chen *et al.* [8] provided an explicit construction to lift a given general random walk P with almost optimal speedup in terms of mixing time. Specifically, they obtained the following result.

Theorem 4 ([8]): For a given random walk P , it is possible to explicitly construct a lifted random walk of P with mixing time $O^*(1/\Phi(P))$. Furthermore, any lifted random walk of P needs at least $\Omega(1/\Phi(P))$ time to mix.

D. Auxiliary Backgrounds

1) ε -Mixing Time: Here we introduce a different (and related) notion of Mixing time which measures more explicitly how fast the random walk converges to the stationarity. The following notions, $\tau(\varepsilon)$, $\tau_2(\varepsilon)$ are related to \mathcal{H} . This relation can be found in detail in the survey by Lovász and Winkler [23]. For example, we will use this relation explicitly in Lemma 8.

Now we define these related definitions of mixing time. To this end, as before, consider a random walk P on a graph $G = (V, E)$. Let $P^t(x, \cdot)$ denote the distribution of the state after t steps under P , starting from an initial state $x \in V$. For the random walk of our interest, $P^t(x, \cdot)$ goes to π as $t \rightarrow \infty$. We present the definitions based on the total variation distance and the χ^2 -distance.

Definition 3 (ε -Mixing Time): Given $\varepsilon > 0$, let $\tau(\varepsilon)$ and $\tau_2(\varepsilon)$ represent ε -Mixing time of the random walk with respect to the total variation distance and the χ^2 -distance respectively. Then, they are

$$\tau(\varepsilon) = \min \left\{ t: \forall x \in V, \frac{1}{2} \sum_{y \in \Omega} |P^t(x, y) - \pi(y)| \leq \varepsilon \right\}$$

$$\tau_2(\varepsilon) = \min \left\{ t: \forall x \in V, \sqrt{\sum_{y \in \Omega} \frac{1}{\pi(y)} (P^t(x, y) - \pi(y))^2} \leq \varepsilon \right\}.$$

2) *Additional Techniques to Bound Mixing Times:* Various techniques have been developed over past three decades or so to estimate Mixing time of a given random walk. The relation between the conductance and the mixing time in the Introduction is one of them. We review some of the key other techniques that will be relevant for this paper.

Fill-up Lemma. Sometimes, due to the difficulty for designing such an exact stopping rule, we use the following strategy for bounding the mixing time \mathcal{H} .

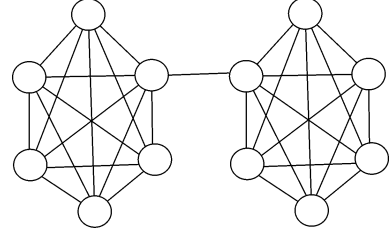


Fig. 2. The Barbell graph with 12 nodes.

Step 1. For a positive constant ε and any starting distribution σ , we design a stopping rule whose stopping distribution γ is ε -far from π (i.e., $\gamma \geq (1 - \varepsilon)\pi$). This gives the upper bound for $H(\sigma, \gamma)$.

Step 2. We bound \mathcal{H} by $H(\sigma, \gamma)$ using the following fact known as *fill-up Lemma* in [1]:

$$\mathcal{H} \leq \frac{1}{1 - \varepsilon} \mathcal{H}_\varepsilon$$

where $\mathcal{H}_\varepsilon = \max_\sigma \min_{\gamma \geq (1 - \varepsilon)\pi} H(\sigma, \gamma)$.

Eigenvalue. If P is reversible, one can view P as a self-adjoint operator on a suitable inner product space and this permits us to use the well-understood spectral theory of self-adjoint operators. It is well known that P has $n = |V|$ real eigenvalues $1 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-1} > -1$. The ε -mixing time $\tau_2(\varepsilon)$ is related as

$$\tau_2(\varepsilon) \leq \left\lceil \frac{1}{\lambda_P} \log \frac{1}{\varepsilon \sqrt{\pi_0}} \right\rceil$$

where $\lambda_P = 1 - \max\{|\lambda_1|, |\lambda_{n-1}|\}$ and $\pi_0 = \min_i \pi_i$. The λ_P is also called the spectral gap of P . When P is nonreversible, we consider PP^* . It is easy to see that the Markov chain with PP^* as its transition matrix is reversible. Let λ_{PP^*} be the spectral gap of this reversible Markov chain. Then, the mixing time of the original Markov chain (with its transition matrix P) is bounded above as

$$\tau_2(\varepsilon) \leq \left\lceil \frac{2}{\lambda_{PP^*}} \log \frac{1}{\varepsilon \sqrt{\pi_0}} \right\rceil. \quad (2)$$

III. PSEUDO-LIFTING

Here our aim is to obtain a random walk with mixing time of the order of the diameter for a given graph G and stationary distribution π . As explained in the Introduction, the following approach based on lifting does not work for this aim: first obtain a random walk with the desired stationary distribution using the Metropolis–Hastings method, and then lift it using the method in [8].

For example, consider the *Barbell graph* G as shown in Fig. 2: *two complete graphs of $n/2$ nodes connected by a single edge.* And, suppose π is uniform. Now, consider a random walk P produced by the Metropolis–Hastings method: the next transition is uniform among all the neighbors for each node. For such a random walk, it is easy to check that $1/\Phi(P) = \Omega(n^2)$ and $\mathcal{H} = \Omega(n^4)$. Therefore, the mixing time of any lifting is at least $\Omega(n^2)$. However, this random walk is ill-designed to begin with because $1/\Phi(P)$ can be decreased up to $O(n)$ by defining its random walk in another way (i.e., increasing the probability of its linkage edge, and adding self-loops to non-linkage nodes not to change its stationary distribution). $1/\Phi(P)$ is still far from

the diameter $D = O(1)$ nevertheless. Hence, from Theorem 4, lifting cannot achieve $O(D)$ -mixing.

Motivated by this limitation, we will use the following new notion of lifting, which we call pseudo-lifting, to design a $O(D)$ -mixing random walk.

Definition 4 (Pseudo-Lifting): A random walk \hat{P} is called a pseudo-lifting of P if there exists a many-to-one function $f : \hat{V} \rightarrow V, T \subset \hat{V}$ with $|T| = |V|$ such that the following holds: (a) for any $\hat{u}, \hat{v} \in \hat{V}, (\hat{u}, \hat{v}) \in \hat{E}$ only if $(f(\hat{u}), f(\hat{v})) \in E$, and (b) for any $u \in V, \hat{\pi}(f^{-1}(u) \cap T) = \frac{1}{2}\pi(u)$.⁸

The property (a) in the definition implies that one can simulate the pseudo-lifting \hat{P} in the original graph G . Furthermore, the property (b) suggests that (by concentrating on the set T), it is possible to simulate the stationary distribution π exactly via pseudo-lifting. Next we present its construction.

A. Construction

For a given random walk P , we will construct the pseudo-lifted random walk \hat{P} of P . It may be assumed that P is given by the Metropolis–Hastings method. We will construct the pseudo-lifted graph \hat{G} by adding vertices and edges to G , and decide the values of the ergodic flows \hat{Q} on \hat{G} , which defines its corresponding random walk \hat{P} .

First, select an arbitrary node v . Now, for each $w \in V$, there exist paths \mathcal{P}_{ww} and \mathcal{P}_{vw} , from w to v and v to w , respectively. We will assume that all the paths are of length D : this can be achieved by repeating same nodes. Now, we construct a pseudo-lifted graph \hat{G} starting from G .

First, create a new node v' which is a copy of the chosen vertex v . Then, for every node w , add directed paths \mathcal{P}'_{ww} , a copy of \mathcal{P}_{ww} , from w to v' . Similarly, add \mathcal{P}'_{vw} (a copy of \mathcal{P}_{vw}) from v' to w . Each addition creates $D - 1$ new interior nodes. Thus, we have essentially created a *virtual star topology* using the paths of the old graph by adding $O(nD)$ new nodes in total. (Every new node is a copy of an old node.)

Now, we define the ergodic flow matrix \hat{Q} for this graph \hat{G} as follows: for an edge (i, j)

$$\hat{Q}_{ij} = \begin{cases} \frac{\delta_1}{2D}\pi_w, & \text{if } (i, j) \in E(\mathcal{P}'_{ww}) \text{ or } E(\mathcal{P}'_{vw}) \\ (1 - \delta_1)Q_{ij}, & \text{if } (i, j) \in E(G) \end{cases}$$

where $\delta_1 \in [0, 1]$ is a constant we will decide later in (3). It is easy to check that $\sum_{ij} \hat{Q}_{ij} = 1, \sum_j \hat{Q}_{ij} = \sum_j \hat{Q}_{ji}$. Hence it defines a random walk on \hat{G} . The stationary distribution of this pseudo-lifting is

$$\hat{\pi}_i = \begin{cases} \frac{\delta_1}{2D}\pi_w, & \text{if } i \in (V(\mathcal{P}'_{ww}) \cup V(\mathcal{P}'_{vw})) \setminus \{w, v'\} \\ (1 - \delta_1 + \frac{\delta_1}{2D})\pi_i, & \text{if } i \in V(G) \\ \frac{\delta_1}{2D}, & \text{if } i = v'. \end{cases}$$

Given the above definition of \hat{Q} and corresponding stationary distribution $\hat{\pi}$, it satisfies the requirements of pseudo-lifting in Definition 4 if we choose δ_1 such that

$$1/2 = \delta_1 \left(1 - \frac{1}{2D}\right) \quad (3)$$

and $T = V(G)$; i.e., T is the set of old nodes.

⁸In fact, $\frac{1}{2}$ can be replaced by any constant between 0 and 1.

B. Mixing Time

We claim the following bound on the mixing time of the pseudo-lifting we constructed.

Theorem 5: The mixing time of the random walk \hat{P} defined by \hat{Q} is $O(D)$.

Proof: We will design a stopping rule where the distribution of the stopping node is $\hat{\pi}$, and analyze its expected length. At first, walk until visiting v' , and toss a coin X with the following probability:

$$X = \begin{cases} 0 & \text{with probability } \frac{\delta_1}{2D} \\ 1 & \text{with probability } \frac{\delta_1(D-1)}{2D} \\ 2 & \text{with probability } 1 - \delta_1 + \frac{\delta_1}{2D} \\ 3 & \text{with probability } \frac{\delta_1(D-1)}{2D}. \end{cases}$$

Depending on the value of X , the stopping node is decided as follows.

- $X = 0$: Stop at v' . The probability for stopping at v' is $\Pr[X = 0] = \frac{\delta_1}{2D}$, which is exactly $\hat{\pi}_{v'}$.
- $X = 1$: Walk a directed path \mathcal{P}'_{vw} , and choose an interior node of \mathcal{P}'_{vw} uniformly at random, and stop there. For a given w , the probability for walking \mathcal{P}'_{vw} is easy to check π_w . There are $D - 1$ many interior nodes, hence, for an interior node i of \mathcal{P}'_{vw} , the probability for stopping at i is
$$\Pr[X = 1] \times \pi_w \times \frac{1}{D - 1} = \frac{\delta_1}{2D}\pi_w = \hat{\pi}_i.$$
- $X = 2$: Stop at the end node w of \mathcal{P}'_{vw} . The probability for stopping at w is
$$\Pr[X = 2] \times \Pr[\text{walk } \mathcal{P}'_{vw}] = \left(1 - \delta_1 + \frac{\delta_1}{2D}\right) \times \pi_w = \hat{\pi}_w.$$
- $X = 3$: Walk until getting a directed path \mathcal{P}'_{ww} , and choose an interior node of \mathcal{P}'_{ww} uniformly at random, and stop there. Until getting a directed path \mathcal{P}'_{ww} , the pseudo-lifted random walk defined by \hat{Q} is the same as the original random walk. Since the distribution $w \in V(G)$ of the walk at the end of the previous step is exactly π , it follows that the distribution π over the nodes of $V(G)$ is preserved under this walk till walking on \mathcal{P}'_{ww} . From the same calculation as for the case $X = 1$, the probability of stopping at the interior node i of \mathcal{P}'_{ww} is $\hat{\pi}_i$.

Therefore, we have established the existence of a stopping rule that takes an arbitrary starting distribution to the stationary distribution $\hat{\pi}$. Now, this stopping rule has an average length $O(D/\delta_1)$: since the probability of getting on a directed path \mathcal{P}'_{ww} at w is $\frac{\delta_1}{2D}/(1 - \delta_1 + \frac{\delta_1}{2D}) = \Theta(\delta_1/D)$, the expected numbers of walks until visiting v' and getting a directed path when $X = 3$ are $O(D/\delta_1) = O(D)$ from (3) in both cases. This completes the proof. \square

IV. PSEUDO-LIFTING: USE OF GEOMETRY

The graph topologies arising in practice, such as those in wireless sensor network deployed in some geographic area or

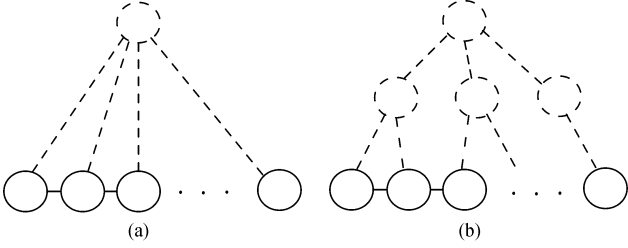


Fig. 3. For a given line graph with n nodes, (a) is the star topology which used in the construction of the pseudo-lifted graph in Section III-A and (b) is the hierarchical star topology which will be used in this section for the new construction of pseudo-lifting.

a nearest neighbor network of unmanned vehicles [30], do possess *geometry* and are far from being expanders. A good model for graphs with geometry is a class of graphs with finite doubling dimension which is defined as follows.

Definition 5 (Doubling Dimension): Consider a metric space $\mathcal{M} = (\mathcal{X}, \mathbf{d})$, where \mathcal{X} is the set of point endowed with a metric \mathbf{d} . Given $x \in \mathcal{X}$, define a ball of radius $r \in \mathbb{R}_+$ around x as $\mathbf{B}(x, r) = \{y \in \mathcal{X} : \mathbf{d}(x, y) < r\}$. Define

$$\rho(x, r) = \inf \left\{ K \in \mathbb{N} : \exists y_1, \dots, y_K \in \mathcal{X}, \right. \\ \left. \mathbf{B}(x, r) \subset \cup_{i=1}^K \mathbf{B}(y_i, r/2) \right\}.$$

Then, the $\rho(\mathcal{M}) = \sup_{x \in \mathcal{X}, r \in \mathbb{R}_+} \rho(x, r)$ is called the *doubling constant* of \mathcal{M} and $\log_2 \rho(\mathcal{M})$ is called the *doubling dimension* of \mathcal{M} . The doubling dimension of a graph $G = (V, E)$ is defined with respect to the metric induced on V by the shortest path metric.

For graphs with finite doubling dimension, we will design a pseudo-lifting with its efficient size. Recall the basic idea for the construction of the pseudo-lifting in Section III is creating a *virtual star topology* using paths from every node to a fixed root, and the length of paths grows the size of the pseudo-lifting. To reduce the overall length of paths, we consider clusters of nodes such that nodes in each cluster are close to each other, and pick a subroot node in each cluster. And then, build a star topology in each cluster around its subroot and connect every subroot to the root. This creates a *hierarchical star topology* (or say a *tree topology*) as you seen in the example of the line graph in Fig. 3(b). Since it needs paths of short length in each cluster, the overall length of paths would be decreased.

For a good clustering, we need to decide which nodes would become subroots. A natural candidate for them is the R -net $Y \subset V$ of a graph G defined as follows.

Definition 6 (R -Net): For a given graph $G = (V, E)$, $Y \subset V$ is an R -net if the following conditions are fulfilled.

- a) For every $v \in V$, there exists $u \in Y$ such that the shortest path distance between u, v is at most R .
- b) The distance between any two $y, z \in Y$ is more than R .

Such an R -net can be found in G greedily, and as will be seen in the proof of Lemma 7, the small *doubling dimension* of G guarantees the existence of a good R -net for our purpose.

A. Construction

For a given random walk P , we will construct the pseudo-lifted random walk \hat{P} of P using a hierarchical star topology. Denote π and $G = (V, E)$ be the stationary distribution and the underlying graph of P again. As the previous construction in Section III-A, we will construct the pseudo-lifted graph \hat{G} by extending G , and define the ergodic flow matrix \hat{Q} on \hat{G} , which leads to its corresponding random walk \hat{P} .

Given an R -net Y , match each node w to the nearest $y \in Y$ (breaking ties arbitrarily). Let $C_y = \{w \mid w \text{ matched to } y\}$ for $y \in Y$. Clearly, $V = \cup_{y \in Y} C_y$. Finally, for each $y \in Y$ and for any $w \in C_y$, we have paths $\mathcal{P}_{wy}, \mathcal{P}_{yw}$ between w and y of length R exactly. Also, for each $y \in Y$, there exist $\mathcal{P}_{yv}, \mathcal{P}_{vy}$ between y and v of length D exactly (we allow the repetition of nodes to hit this length exactly).

Now, we construct the pseudo-lifted graph \hat{G} . As the construction in Section III-A, select an arbitrary node $v \in V$ and create its copy v' again. Further, for each $y \in Y$, create two copies y'_1 and y'_2 . Now, add directed paths \mathcal{P}'_{wy} , a copy of \mathcal{P}_{wy} , from w to y'_1 and add \mathcal{P}'_{yv} , a copy of \mathcal{P}_{yv} , from y'_1 to v' . Similarly, add \mathcal{P}'_{vy} and \mathcal{P}'_{yw} between v', y'_2 and y'_2, w . In total, this construction for \hat{G} adds $2D|Y| + 2Rn$ edges to G . Now, the ergodic flow matrix \hat{Q} on \hat{G} is defined as follows: for any (i, j) of \hat{G}

$$\hat{Q}_{ij} = \begin{cases} \frac{\delta_2}{2(R+D)} \pi_w, & \text{if } (i, j) \in E(\mathcal{P}'_{wy}) \text{ or } E(\mathcal{P}'_{yw}) \\ \frac{\delta_2}{2(R+D)} \pi(C_y), & \text{if } (i, j) \in E(\mathcal{P}'_{yv}) \text{ or } E(\mathcal{P}'_{vy}) \\ (1 - \delta_2) Q_{ij}, & \text{if } (i, j) \in E(G) \end{cases}$$

where $\pi(C_y) = \sum_{w \in C_y} \pi_w$ and $\delta_2 \in [0, 1]$ is a constant decided later.⁹ It can be checked that $\sum_{ij} \hat{Q}_{ij} = 1, \sum_j \hat{Q}_{ij} = \sum_j \hat{Q}_{ji}$. Hence it defines a random walk on \hat{G} . The stationary distribution of this pseudo-lifted chain is given at the bottom of the page. To guarantee that this chain is indeed the pseudo-lifting of the original random walk P , consider $T = V(G)$ and δ_2 , where

$$\frac{1}{2} = \delta_2 \left(1 - \frac{1}{2(R+D)} \right). \quad (4)$$

Note that \hat{G} has exactly $|E| + 2Rn + 2D|Y|$ edges.

⁹See (4) and check $\delta_2 \approx 1/2$.

$$\hat{\pi}_i = \begin{cases} \frac{\delta_2}{2(R+D)} \pi_w, & \text{if } i \in (V(\mathcal{P}'_{wy}) \cup V(\mathcal{P}'_{yw})) \setminus \{w, y'_1, y'_2\} \\ \frac{\delta_2}{2(R+D)} \pi(C_y), & \text{if } i \in (V(\mathcal{P}'_{yv}) \cup V(\mathcal{P}'_{vy})) \setminus \{v'\} \\ \left(1 - \delta_2 \left(1 - \frac{\delta_2}{2(R+D)} \right) \right) \pi_i, & \text{if } i \in V(G) \\ \frac{\delta_2}{2(R+D)}, & \text{if } i = v'. \end{cases}$$

B. Mixing Time and Size: Proof of Theorem 1

We prove two lemmas about the performance of pseudo-lifting we constructed, and they imply Theorem 1. At first, we state the following result about its mixing time, and the proof can be done similarly as the proof of Theorem 5.

Lemma 6: The mixing time of the random walk \hat{P} defined by \hat{Q} is $O(D)$.

Proof: Consider the following stopping rule. Walk until visiting v' , and toss a coin X with the following probability.

$$X = \begin{cases} 0 & \text{with probability } \frac{\delta_2}{2(R+D)} \\ 1 & \text{with probability } \frac{\delta_2 D}{2(R+D)} \\ 2 & \text{with probability } \frac{\delta_2(R-1)}{2(R+D)} \\ 3 & \text{with probability } 1 - \delta_2 \left(1 - \frac{\delta_2}{2(R+D)}\right) \\ 4 & \text{with probability } \frac{\delta_2(R-1)}{2(R+D)} \\ 5 & \text{with probability } \frac{\delta_2 D}{2(R+D)}. \end{cases}$$

Depending on the value of X

- $X = 0$: Stop at v' .
- $X = 1$: Walk on a directed path \mathcal{P}'_{vy} , and choose its interior node uniformly at random, and stop there.
- $X = 2$: Walk until getting a directed path \mathcal{P}'_{yw} , and choose its interior node uniformly at random, and stop there.
- $X = 3$: Walk until getting an old node in $V(G)$, and stop there.
- $X = 4$: Walk until getting a directed path \mathcal{P}'_{wy} , and choose its interior node uniformly at random, and stop there.
- $X = 5$: Walk until getting a directed path \mathcal{P}'_{yv} , and choose its interior node uniformly at random, and stop there.

It can be checked, using arguments similar to that in proof of Theorem 5, that the distribution of the stopped node is precisely $\hat{\pi}$. Also, we can show that the expected length of this stopping rule is $O(\frac{R+D}{\delta_2}) = O(\frac{D}{\delta_2}) = O(D)$ from (4). This is primarily true because the probability of getting on a directed path \mathcal{P}'_{wy} at w is $\Theta(\delta_2/(R+D))$. \square

Now we apply the hierarchical construction to the case of graphs with constant doubling dimension, and show the guarantee for the size of the pseudo-lifting in terms of its doubling dimension.

Lemma 7: Given a graph G with a constant doubling dimension ρ and its diameter D , the hierarchical construction gives a pseudo-lifted graph \hat{G} with its size $|\hat{E}| = O(Dn^{1-\frac{1}{\rho+1}})$.

Proof: The property of doubling dimension graph implies that there exists an R -net Y such that $|Y| \leq (2D/R)^\rho$ (cf. [2]). Consider $R = D2^{\frac{\rho}{\rho+1}}n^{-\frac{1}{\rho+1}}$. This is an appropriate choice because

$$R = D2^{\frac{\rho}{\rho+1}}n^{-\frac{1}{\rho+1}} > Dn^{-\frac{1}{\rho+1}} > n^{\frac{1}{\rho}-\frac{1}{\rho+1}} > 1$$

(the second inequality is from $n \leq D^\rho$). Given this, the size of the pseudo-lifted graph \hat{G} is

$$\begin{aligned} |\hat{E}| &= |E| + 2Rn + 2D|Y| \\ &\leq |E| + 2D \left(\frac{2^{\frac{\rho}{\rho+1}}}{n^{\frac{1}{\rho+1}}}\right)n + 2D \left(2\frac{n^{\frac{1}{\rho+1}}}{2^{\frac{\rho}{\rho+1}}}\right)^\rho \\ &= |E| + O\left(Dn^{1-\frac{1}{\rho+1}}\right). \end{aligned}$$

Since $|E| = O(n)$ and $D = \Omega(n^{1/\rho})$, we have that $|\hat{E}| = O(Dn^{1-\frac{1}{\rho+1}})$. \square

V. APPLICATION: BACK TO AVERAGING

As we stated in the Introduction, consider the following computation problem of the distributed averaging. Given a connected network graph $G = (V, E)$, where $V = \{1, 2, \dots, n\}$, each node $i \in V$ has a value $x_i \in \mathbb{R}$. Then the goal is to compute the average of $\mathbf{x} = [x_i]$ only by communications between adjacent nodes

$$x_{\text{ave}} = \frac{1}{n} \sum_i x_i. \quad (5)$$

This problem arises in many applications such as distributed estimation [31], distributed spectral decomposition [18], estimation and distributed data fusion on *ad hoc* networks [24], distributed subgradient method for eigenvalue maximization [6], inference in Gaussian graphical models [25], and coordination of autonomous agents [16].

A. Linear Iterative Algorithm

A popular and quite simple approach for this computation is a method based on linear iterations [32] as follows. Suppose we are given with a graph conformant random walk P which has the uniform stationary distribution π i.e., $\pi^T P = \pi^T$. The linear iteration algorithm is described as follows. At time t , each node $i \in V$ has an estimate $y_i(t)$ of x_{ave} and initially $y_i(0) = x_i$. At time $t = 1, 2, \dots$ for each edge (i, j) of G , node i sends value $P_{ji}y_i(t)$ to node j . Then each node j sums up the values received as its estimate at time $t + 1$, that is

$$y_j(t+1) = \sum_{i=1}^n P_{ji}y_i(t).$$

Under the condition that P is ergodic, i.e., P is connected and aperiodic, it is known that [32]

$$\lim_{t \rightarrow \infty} y(t) = \lim_{t \rightarrow \infty} P^t \mathbf{x} = \left(\sum_i x_i \right) \pi = \frac{1}{n} \sum_i x_i \mathbf{1} = x_{\text{ave}} \mathbf{1},$$

where $\mathbf{1} = [1]$.

Specifically, as we already saw in the Introduction, ε -computation time $T_\varepsilon(P)$ is defined as

$$T_\varepsilon(P) = \inf \left\{ t : \forall \mathbf{x} \in \mathbb{R}_+^n, \frac{\|P^t \mathbf{x} - x_{\text{ave}} \mathbf{1}\|_\infty}{x_{\text{ave}}} \leq \varepsilon \right\}. \quad (6)$$

The quantity $T_\varepsilon(P)$ is well known to be related to the mixing time $\mathcal{H}(P)$. More precisely, we prove Lemma 8, which implies

$$T_\varepsilon(P) = O^* \left(\mathcal{H}(P) \log \frac{1}{\varepsilon} \right). \quad (7)$$

Since each edge (i, j) such that $P_{ij} > 0$ performs an exchange of values per each iteration, the number of operations performed per iteration across the network is at most $|E|$. Thus, the total number of operations of the linear iterations to obtain the approximation of x_{ave} scales like

$$C_\varepsilon(P) := T_\varepsilon(P) \times |E|. \quad (8)$$

Therefore, the task of designing an appropriate P with small $\mathcal{H}(P)$ is important to minimize both $T_\varepsilon(P)$ and $C_\varepsilon(P)$.

B. Linear Iterative Algorithm With Pseudo-Lifting: Proof of Theorem 2

We present a linear iterative algorithm that utilizes the pseudo-lifted version of a given matrix P on the original graph G . The main idea behind this implementation is to run the standard linear iterations in $\hat{G} = (\hat{V}, \hat{E})$ with the pseudo-lifted chain \hat{P} . However, we wish to implement this on $G = (V, E)$ and not \hat{G} . Now recall that \hat{G} has the following properties: (a) each node $\hat{v} \in \hat{V}$ is a copy of a node $v \in V$, and (b) each edge (\hat{u}, \hat{v}) is a copy of edge $(u, v) \in E$, where \hat{u}, \hat{v} are copies of $u, v \in V$, respectively. Therefore, each node $\hat{v} \in \hat{V}$ can be simulated by a node $v \in V$ where \hat{v} is a copy of v for the purpose of linear iterations. Thus, it is indeed possible to simulate the pseudo-lifted version of a matrix P on G by running multiple threads (in the language of the computer programming) on each node of G . We state this approach formally as follows:

1. Given graph $G = (V, E)$, we wish to compute the average x_{ave} at all nodes. For this, first produce a matrix P using the Metropolis–Hastings method with the uniform stationary distribution.
2. Construct the pseudo-lifting \hat{P} based on P as explained in Section IV. This pseudo-lifted random walk has a stationary distribution $\hat{\pi}$ on a graph \hat{G} .
3. As explained below, implement the linear iterative algorithm based on \hat{P} on the original graph G .
 - Let t be the index of iterations of the algorithm and initially it be equal to 0.
 - For each node $\hat{v} \in \hat{V}$, maintain a number $y_{\hat{v}}(t)$ at the t th iteration. This is maintained at the node $v \in V$ where \hat{v} is a copy of v . The initialization of these values is stated as follows.
 - Recall that, \hat{V} contains V as its subset. Recall that they are denoted as $V(G) \subset \hat{V}$, and each $v \in G$ has its copy $\hat{v} \in \hat{V}(G)$.
 - For each $\hat{v} \in V(G)$, initialize $y_{\hat{v}}(0) = x_v$.
 - For each $\hat{v} \in \hat{V} \setminus V(G)$, initialize $y_{\hat{v}}(0) = 0$.

- In the $t + 1$ th iteration, update

$$y_{\hat{v}}(t+1) = \sum_{\hat{u} \in \hat{V}} \hat{P}_{\hat{v}\hat{u}} y_{\hat{u}}(t).$$

This update is performed by each node v through receiving information from its neighbors u in G , where \hat{v} is a copy of v and neighbors (of \hat{v}) \hat{u} are copies of neighbors (of v) u .

4. At the end of the t th iteration, each node v produces its estimate as $2y_{\hat{v}}(t)$, $\hat{v} \in \hat{V}(G)$.

It can be easily verified that since the above algorithm is indeed implementing the linear iterative algorithm based on \hat{P} , the ε

computation time is $T_\varepsilon(\hat{P})$ and the total number of communications performed is $C_\varepsilon(\hat{P})$. In what follows, for the completeness we bound $T_\varepsilon(\hat{P})$ and $C_\varepsilon(\hat{P})$.

Lemma 8: $T_\varepsilon(\hat{P}) = O(\mathcal{H}(\hat{P}) \log \frac{1}{\varepsilon \pi_0})$.

Proof: Here, we need the ε -mixing time $\tau(\varepsilon)$ based on the total variance distance, and recall its definition in Section II-D

$$\tau(\varepsilon) = \min \left\{ t : \forall i \in G, \frac{1}{2} \sum_{j \in G} |P_{ij}^t - \pi_j| \leq \varepsilon \right\}.$$

The following relation between two different mixing time $\tau(\varepsilon)$ and \mathcal{H} is known (see [23]):

$$\tau(\varepsilon) = O\left(\mathcal{H} \log \frac{1}{\varepsilon}\right).$$

If t is larger than $\tau(\varepsilon \pi_0 / 4)$ of \hat{P} , which is $O(\mathcal{H}(\hat{P}) \log \frac{1}{\varepsilon \pi_0 / 4})$

$$\begin{aligned} & |y_i(t) - \langle y(0), \hat{\pi} \rangle| \\ &= \left| \sum_j \hat{P}_{ij}^t y_j(0) - \sum_j y_j(0) \hat{\pi}_j \right| \\ &\leq \sum_j y_j(0) \left| \hat{P}_{ij}^t - \hat{\pi}_j \right| \\ &\stackrel{(a)}{\leq} \sum_j y_j(0) \frac{\varepsilon \pi_0}{2} \stackrel{(b)}{\leq} \sum_j y_j(0) \varepsilon \hat{\pi}_j = \varepsilon \langle y(0), \hat{\pi} \rangle \end{aligned}$$

where (a) is from $|\hat{P}_{ij}^t - \hat{\pi}_j| \leq \sum_j |\hat{P}_{ij}^t - \hat{\pi}_j| \leq 2 \times \frac{\varepsilon \pi_0}{4} = \frac{\varepsilon \pi_0}{2}$, and (b) is because $\hat{\pi}_j > \frac{1}{2} \pi_j \geq \frac{1}{2} \pi_0$ for every old node $j \in V(G)$, and $y_j(0) = 0$ otherwise. This completes the proof. \square

From the proof of Lemma 8, note that the relation $T_\varepsilon(P) = O(\mathcal{H}(P) \log \frac{1}{\varepsilon \pi_0})$ holds for any random walk P . Therefore, $T_\varepsilon(\hat{P}) = O(D \log \frac{1}{\varepsilon \pi_0})$ and $C_\varepsilon(\hat{P}) = T_\varepsilon(\hat{P}) \times |\hat{E}| = O(D^2 n^{1-\frac{1}{d+1}} \log \frac{1}{\varepsilon \pi_0})$ since $\mathcal{H}(\hat{P}) = O(D)$ and $|\hat{E}| = O(D n^{1-\frac{1}{d+1}})$ from Lemmas 6 and 7. This also completes the proof of Theorem 2.

C. Comparison With Other Algorithms

Even considering any possible algorithms based on passing messages, the lower bound of the performance guarantees in the averaging problem is $O(D)$ for the running time, and $O(Dn)$ for the total number of operations. Therefore, our algorithm using pseudo-lifting gives the best running time, and possibly loses the $\frac{O^*(D^2 n^{1-\frac{1}{d+1}})}{O(Dn)} = O^*(D/n^{\frac{1}{d+1}})$ factor in terms of the total number of operations compared to the best algorithm. For example, when G is a d -dimensional grid graph, this loss is only $O^*(D/n^{\frac{1}{d+1}}) = O^*(n^{1/d}/n^{\frac{1}{d+1}}) = O^*(n^{\frac{1}{d(d+1)}})$ since the doubling dimension of G is d and its diameter D is $O(n^{1/d})$. The standard linear iterations using the Metropolis–Hastings method loses the $\Omega(n^{1/d})$ factor in both the running time and the total number of operations (see Table I).

We take note of the following subtle matter: the nonreversibility is captured in the transition probabilities of the underlying Markov chain (or random walk); but the linear iterative algorithm does not change its form other than this detail.

TABLE I
COMPARISON OF PSEUDO-LIFTING WITH THE METROPOLIS–HASTINGS METHOD.
HERE, WE ASSUME G HAS $\Theta(n)$ EDGES

	Metropolis-Hastings	Pseudo-Lifting	Optimal
Mixing time(Running time) : d -dim. grid graph	$\Omega\left(\frac{1}{\Phi^2(P)}\right)$: $O^*\left(n^{\frac{2}{d}}\right)$	$O(D)$: $O\left(n^{\frac{1}{d}}\right)$	D : $n^{\frac{1}{d}}$
Size (dbl. dim. ρ) : d -dim. grid graph	$\Theta(n)$: $\Theta(n)$	$O\left(n^{\frac{\rho}{\rho+1}}D\right)$: $O\left(n^{1+\frac{1}{d(d+1)}}\right)$	n : n
Total # of operations : d -dim. grid graph	$\Omega\left(\frac{n}{\Phi^2(P)}\right)$: $O^*\left(n^{1+\frac{2}{d}}\right)$	$O\left(n^{\frac{\rho}{\rho+1}}D^2\right)$: $O^*\left(n^{1+\frac{d+2}{d(d+1)}}\right)$	nD : $n^{1+\frac{1}{d}}$

VI. LIFTING USING EXPANDERS

We introduced the new notion of pseudo-lifting for the applications of interest, one of which was the distributed averaging. However, since it may not be relevant to certain other applications, we optimize the size of lifting (not pseudo-lifting) in [8]. The basic motivation of our construction is using the expander graph, instead of the complete graph in [8], to reduce the size of the lifting.

A. Preliminaries

In what follows, we will consider only P such that $P \geq I/2$. This is without loss of generality due to the following reason. Suppose such is not the case, then we can modify it as $(I+P)/2$; the mixing time of $(I+P)/2$ is within a constant factor of the mixing time of P .

1) *Multi-Commodity Flows*: In [8], the authors use a multi-commodity flow to construct a specific lifting of a given random walk P to speed up its mixing time. Specifically, they consider a multi-commodity flow problem on G with the capacity constraint on edge $(u, v) \in E$ given by Q_{uv} . A flow from a source s to a destination t , denoted by f , is defined as a nonnegative function on edges of G so that

$$\sum_j f(ji) = \sum_j f(ij)$$

for every node $i \neq s, t$. The value of the flow is defined by

$$\begin{aligned} \text{val}(f) &= \sum_j f(sj) - \sum_j f(js) \\ &= \sum_j f(jt) - \sum_j f(tj), \end{aligned}$$

and the cost of flow f^{st} is defined as

$$\text{cost}(f) = \sum_{(i,j) \in E} f(ij).$$

A *multi-commodity flow* is a collection $f = (f^{st})$ of flows, where each f^{st} is a flow from s to t . Define the *congestion* of a multi-commodity flow f as

$$\max_{(i,j) \in E} \frac{\sum_{s,t} f^{st}(ij)}{Q_{ij}}.$$

Consider the following optimization problem, essentially trying to minimize the congestion and the cost simultaneously under the condition for the amount of flows:

$$\begin{aligned} &\text{minimize } K \\ &\text{subject to } \text{val}(f^{st}) = \pi_s \pi_t, \quad \forall s, t, \\ &\quad \sum_{s,t} f^{st}(ij) \leq K Q_{ij}, \quad \forall (i, j) \in E, \\ &\quad \sum_t \text{cost}(f^{st}) \leq K \pi_s, \quad \sum_s \text{cost}(f^{st}) \leq K \pi_t, \quad \forall s, t. \end{aligned}$$

Let C be the optimal solution of the above problem. It is easy to see that $C \geq 1/\Phi$. Further, if P is reversible, then result of Leighton and Rao [22] on the approximate multi-commodity implies that

$$C = O\left(\frac{1}{\Phi} \log \frac{1}{\pi_0}\right).$$

Let the optimal multi-commodity flow of the above problem be F_1 , and we can think of F_1 as a weighted collection of directed paths. In [8], the authors modified F_1 , and got a new multi-commodity flow F_2 that has the same amount of $s-t$ flows as F_1 , while its congestion and path length are at most $12C$. They used F_2 to construct a lifting \hat{P} with mixing time \hat{H} such that

$$\hat{H} \leq 144C.$$

Also, they showed that the mixing time of any lifting \hat{P} is greater than $C/2$, hence their lifted Markov chain has almost optimal speedup within a constant factor.

To obtain a lifting with the smaller size than that in [8], we will study the existence of the specific k -commodity flow with short path lengths. For this, we will use a *balanced multi-commodity flow*, which is a multi-commodity flow with the following condition for the amount of flows:

$$\text{val}(f^{st}) = g(s, t), \quad \forall s, t$$

and $g(s, t)$ satisfies the balanced condition

$$\sum_t g(s, t) \leq \pi_s, \quad \sum_s g(s, t) \leq \pi_t, \quad \forall s, t.$$

Therefore, F_1 and F_2 are also balanced multi-commodity flows with $g(s, t) = \pi_s \pi_t$. Given a multi-commodity flow f , let $C(f)$

be its congestion and $D(f)$ be the length of the longest flow-path. Then, the flow number T is defined as follows:

$$T = \min_f (\max\{C(f), D(f)\})$$

where the minimum is taken over all balanced multi-commodity flows with $g(s, t) = \pi_s \pi_t$. Hence, F_2 implies $T \leq 12C$. The following claim appears in [20].

Claim 9 (Claim 2.2 in [20]): For any $g(s, t)$ satisfying the balanced condition (not necessarily $g(s, t) = \pi_s \pi_t$), there exists a balanced multi-commodity flow f with $g(s, t)$ such that $\max\{C(f), D(f)\} \leq 2T$.

2) *Expanders:* The expander graphs are sparse graphs which have high connectivity properties, quantified using the edge expansion $h(G)$ as defined as

$$h(G) = \min_{1 \leq |S| \leq \frac{n}{2}} \frac{|\partial(S)|}{|S|}$$

where $\partial(S)$ is the set of edges with exactly one endpoint in S . For constants d and c , a family $\mathcal{G} = \{G_1, G_2, \dots\}$ of d -regular graphs is called a (d, c) -expander family if $h(G) > c$ for every $G \in \mathcal{G}$. There are many explicit constructions of a (d, c) -expander family available in recent times. We will use a (d, c) -expander graph $G^{Ex} = (V, E^{Ex})$ (i.e., $V^{Ex} = V$), and a transition matrix P^{Ex} defined on this graph. For a given π , we can define a reversible P^{Ex} so that its stationary distribution is π as follows:

$$P_{ij}^{Ex} = \begin{cases} \frac{\pi_0}{d\pi_i}, & \text{if } (i, j) \in E^{Ex} \\ 1 - \frac{\pi_0}{\pi_i}, & \text{if } i = j. \end{cases}$$

In the case of $\pi_{\max} = O(\pi_0)$, it is easy to check that $\Phi(P^{Ex}) = \Theta(h(G)) = \Omega(1)$, where $\Phi(P^{Ex})$ is the conductance of P^{Ex} . Hence, $\lambda_{P^{Ex}} = \Omega(1)$, and the random walk defined by P^{Ex} mixes fast. In this section, we will consider only such π .

B. Construction

We use the multi-commodity flow based construction which was introduced by Dasgupta and Freund in [8]. They essentially use a multi-commodity flow between source–destination pairs for all $s, t \in V$. Instead, we will use a balanced multi-commodity flow between source–destination pairs that are obtained from an expander. Thus, the essential change in our construction is the use of an expander in place of a complete graph used in [8]. A caricature of this lifting is explained in Fig. 4. However, this change makes the analysis of the mixing time a lot more challenging and requires us to use different analysis techniques. Further, we use arguments based on the classical linear programming to derive the bound on the size of lifting.

To this end, we consider the following multi-commodity flow: let $G^{Ex} = (V, E^{Ex})$ be an expander with a transition matrix P^{Ex} and a stationary distribution π as required—this is feasible since we have assumed $\pi_{\max} = O(\pi_0)$. We note that this assumption is used only for the existence of expanders. Consider a multi-commodity flow $f = (f^{st})_{(s,t) \in E^{Ex}}$ so that

- $\text{val}(f^{st}) = \pi_s P_{st}^{Ex} = Q_{st}^{Ex}, \forall (s, t) \in E^{Ex};$
- $\sum_{s,t} f^{st}(ij) \leq K Q_{ij}, \forall (i, j) \in E;$

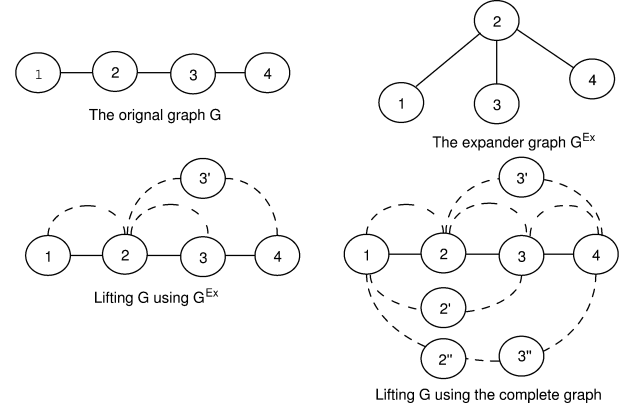


Fig. 4. A caricature of lifting using expander. Let line graph G be a line graph with four nodes. We wish to use an expander G^{Ex} with four nodes, shown on the top-right side of the figure. G is lifted by adding paths that correspond to edges of expander. For example, an edge $(2, 4)$ of expander is added as path $(2, 3', 4)$. We also draw the lifting in [8] which uses the complete graph.

Lemma 10: There is a feasible multi-commodity flow in the above flow problem with congestion (K) and path-length at most W , where $W = O^*(1/\Phi(P))$.

Proof: The conclusion is derived directly from Claim 9 since the flow number T is less than $12C = O^*(1/\Phi(P))$ and the flow considered is a balanced multi-commodity flow i.e., $W = 24C = O^*(1/\Phi(P))$. \square

Now, we can think of this multi-commodity flow as a weighted collection of directed paths $\{(\mathcal{P}_r, w_r) : 1 \leq r \leq N\}$, where the total weight of paths from node s to t is $\pi_s P_{st}^{Ex}$, where $(s, t) \in E^{Ex}$. Let ℓ_r be the length of path \mathcal{P}_r . From Lemma 10, we have the following:

$$\sum_r w_r = 1, \quad \ell_r \leq W \quad (9)$$

$$\sum_{r: \mathcal{P}_r \text{ starts at } i} w_r = \pi_i, \quad \sum_{r: \mathcal{P}_r \text{ ends at } i} w_r = \pi_i, \quad \text{for } i \in V \quad (10)$$

$$\sum_{r: (i,j) \in E(\mathcal{P}_r)} w_r \leq W Q_{ij}, \quad \text{for } (i, j) \in E. \quad (11)$$

Using such a collection of weighted paths, we construct the desired lifting next. As in Fig. 4, we construct the lifted graph $\hat{G} = (\hat{V}, \hat{E})$ from G by adding a directed path \mathcal{P}'_r of length ℓ_r connecting i to j if \mathcal{P}_r goes from i to j . Subsequently, $\ell_r - 1$ new nodes are added to the original graph. The ergodic flow on an edge (i, j) of the lifted chain is defined by

$$\hat{Q}_{ij} = \begin{cases} w_r/2W, & \text{if } (i, j) \in E(\mathcal{P}'_r) \\ Q_{ij} - \sum_{r: ij \in E(\mathcal{P}_r)} w_r/2W, & \text{if } (i, j) \in E(G). \end{cases}$$

It is easy to check that it defines a Markov chain on \hat{G} , and a natural way of mapping the paths \mathcal{P}'_r onto the paths \mathcal{P}_r collapses the random walk on \hat{G} onto the random walk on G . The stationary distribution of the lifted chain is

$$\hat{\pi}_i = \begin{cases} w_r/2W, & \text{if } i \in V(\mathcal{P}'_r) \setminus V(G) \\ \pi_i - \sum_{r: \mathcal{P}_r \text{ thru } i} w_r/2W, & \text{if } i \in V(G). \end{cases}$$

Thus, the above stated construction is a valid lifting of the given Markov chain P defined on G .

C. Mixing Time and Size: Proof of Theorem 3

We prove two lemmas about the performance of lifting we constructed, and they imply Theorem 3. At first, we state and prove the lemma which bounds the mixing time of the lifted chain we constructed.

Lemma 11: The mixing time $\hat{\mathcal{H}}$ of the lifted Markov chain represented by \hat{Q} defined on \hat{G} is $O^*(1/\Phi(P))$.¹⁰

Proof: By the property of expanders, we have $\lambda_{PE_x} = \Omega(1)$. Therefore, it is sufficient to show that

$$\hat{\mathcal{H}} = O\left(\frac{W}{\lambda_{PE_x}} \log \frac{1}{\pi_0}\right).$$

First, note that for any node $i \in V$ (i.e., an original node i in G)

$$\frac{1}{2}\pi_i \leq \hat{\pi}_i \leq \pi_i. \quad (12)$$

Now, under the lifted Markov chain, the probability of getting on any directed path \mathcal{P}'_r starting at i is

$$\hat{P}_{ij} = \frac{\hat{Q}_{ij}}{\hat{\pi}_i} = \frac{w_r}{2W\hat{\pi}_i}.$$

Hence, the probability of getting on any directed path starting at i is

$$\sum_{r:\mathcal{P}'_r \text{ starts at } i} \frac{w_r}{2W\hat{\pi}_i} = \frac{1}{2W\hat{\pi}_i} \sum_{r:\mathcal{P}'_r \text{ starts at } i} w_r = \frac{\pi_i}{2W\hat{\pi}_i}.$$

From (12), this is bounded between $\frac{1}{2W}$, and $\frac{1}{W}$.

To study the $\hat{\mathcal{H}}$, we will focus on the induced random walk (or Markov chain) on original nodes $V \subset \hat{V}$ by the lifted Markov chain \hat{P} . Let \hat{P}^V be the transition matrix of this induced random walk. Then

$$\hat{P}_{ij}^V = \hat{P}_{ij} + \sum_{r:\mathcal{P}'_r \text{ goes from } i \text{ to } j} \frac{w_r}{2W\hat{\pi}_i}.$$

Now, $\hat{P}^V \geq \hat{P} \geq I/4$, because $\hat{P}_{ii} = \hat{Q}_{ii}/\hat{\pi}_i \geq Q_{ii}/2\hat{\pi}_i = P_{ii}\pi_i/2\hat{\pi}_i \geq P_{ii}/2 \geq I/4$. Here we have assumed that $P \geq I/2$ as discussed earlier. Now

$$\begin{aligned} \hat{P}_{ij}^V &\geq \frac{1}{2W\hat{\pi}_i} \sum_{r:\mathcal{P}'_r \text{ goes from } i \text{ to } j} w_r \\ &= \frac{\pi_i P_{ij}^{Ex}}{2W\hat{\pi}_i} \geq \frac{1}{2W} P_{ij}^{Ex}. \end{aligned}$$

Also, its stationary distribution $\hat{\pi}^V$ is $\hat{\pi}_i^V = \frac{\hat{\pi}_i}{\hat{\pi}(V)}$. Therefore, by (12) we have $\frac{1}{2}\pi_i \leq \hat{\pi}_i^V \leq 2\pi_i$. Now, we can apply Claim 14 to obtain the following:

$$\lambda_{\hat{P}^V(\hat{P}^V)^*} = \Omega\left(\frac{1}{W} \lambda_{PE_x}\right). \quad (13)$$

We are ready now to design the following stopping rule Γ that will imply that the desired bound on $\hat{\mathcal{H}}$.

¹⁰The precise bound is $O(W \log \frac{1}{\pi_0})$.

- (i) Walk until visiting old nodes of $V \subset \hat{V}$ for T times, where $T := \lceil 2 \log(2/\hat{\pi}_0^V) / \lambda_{\hat{P}^V(\hat{P}^V)^*} \rceil$. Let this T^{th} old node be denoted by X .
- (ii) Stop at X with probability $1/2$.
- (iii) Otherwise, continue walking until getting onto any directed path \mathcal{P}'_r ; choose an interior node Y of \mathcal{P}'_r uniformly at random and stop at Y .

From the relation (2) in Section II-D with $\varepsilon = \frac{1}{2}\sqrt{\hat{\pi}_0^V}$, it follows that after time T as defined above the Markov chain \hat{P}^V , restricted to old nodes V , has distribution close to $\hat{\pi}^V$ i.e.,

$$|\Pr(X = w) - \hat{\pi}_w^V| \leq \hat{\pi}_w^V/2, \quad \forall w \in V.$$

According to the above stopping rule, we stop at an old node w with probability $1/2$. Therefore, for any $w \in V$, we have that the stopping time Γ stops at w with probability at least $\hat{\pi}_w^V/4 \geq \pi_w/8 \geq \hat{\pi}_w/8$. With probability $1/2$, the rule does not stop at the node X . Let w^k be the k th point in the walk starting from X . Because at any old node i , the probability of getting on any directed path is between $\frac{1}{2W}$ and $\frac{1}{W}$, a coupling argument shows that for any old node i

$$\Pr(w^k = i | w^0, \dots, w^k \text{ are old nodes}) \geq \left(1 - \frac{1}{W}\right)^k \frac{1}{2} \hat{\pi}_i^V.$$

If w is a new point on the directed path \mathcal{P}'_r which connects the old node i to j . Then

$$\begin{aligned} \Pr(\Gamma \text{ stop at } w) &\geq \frac{1}{2} \sum_{k=0}^{\infty} \text{Prob}(w^k = i | w^0, \dots, w^k \text{ are old points}) \\ &\quad \times \text{Prob}(\text{at } i, \text{ get on the path } \mathcal{P}'_r) \times \frac{1}{\ell_r} \\ &\geq \frac{1}{2} \sum_{k=0}^{\infty} \left(1 - \frac{1}{W}\right)^k \frac{1}{2} \hat{\pi}_i^V \frac{w_r}{2W\hat{\pi}_i} \frac{1}{W} \\ &\geq \frac{w_r}{16W^2} \sum_{k=0}^{\infty} \left(1 - \frac{1}{W}\right)^k \\ &= \frac{w_r}{16W} \\ &= \frac{1}{8} \hat{\pi}_w. \end{aligned}$$

The average length of this stopping rule is $O(T + W)$. By (13)

$$\begin{aligned} O(T + W) &= O\left(\left\lceil \frac{2}{\lambda_{\hat{P}^V(\hat{P}^V)^*}} \log(2/\pi_0) \right\rceil + W\right) \\ &= O\left(\frac{W}{\lambda_{PE_x}} \log(1/\pi_0)\right). \end{aligned}$$

Thus, we have established that the stopping rule Γ has the average length $O(W \log 1/\pi_0)$ and the distribution of the stopping node is $\Omega(\hat{\pi})$. Therefore, using the *fill-up* lemma stated in [1], it follows that $\hat{\mathcal{H}} = O(W \log 1/\pi_0)$. \square

Also, we bound the size of the lifted chain we constructed as follows.

Lemma 12: The size of the lifted Markov chain can be bounded above as $O^*(|E|/\Phi(P))$.¹¹

Proof: We want to establish that the size of the lifted chain in terms of the number of edges, i.e., $|\hat{E}| = O^*(|E|/\Phi(P))$. Note that, the lifted graph \hat{G} is obtained by adding paths that appeared in the solution of the multi-commodity flow problem. Therefore, to establish the desired bound we need to establish a bound on the number of distinct paths as well as their lengths.

To this end, let us reformulate the multi-commodity flow based on expander G^{Ex} as follows. For each $(s, t) \in E^{Ex}$, we add a flow between s and t . Let this flow be routed along possibly multiple paths. Let P_{stj} denote the j th path from s to t and x_{stj} be the amount of flow sent along this path. The length ℓ_{stj} of P_{stj} is at most W as the discussion in Lemma 10. Let the overall solution, denoted by $\{(P_r, w_r)\}$, give a feasible solution in the following polytope with x_{stj} as its variables:

$$\begin{aligned} \sum_j x_{stj} &= \pi_s P_{st}^{Ex}, \quad \forall (s, t) \in E^{Ex} \\ \sum_{st \in E^{Ex}} \sum_{j: e \in P_{stj}} x_{stj} &\leq W Q_e, \quad \forall e \in E \\ x_{stj} &\geq 0 \quad \forall s, t, j. \end{aligned}$$

Clearly, any feasible solution in this polytope, say $\{(P_r, w_r)\}$, will work for our lifting construction. Now, the size of its support set is $|\{(P_r, w_r)\}|$. If we consider the extreme point of this polytope, the size of its support set is at most $|E^{Ex}| + |E| = O(|E|)$ because the extreme point is a unique solution of a sub-collection of linear constraints in this polytope. Hence, if we choose such an extreme point $\{(P_r, w_r)\}$ for our lifting, the size of our lifted chain $|\hat{E}|$ is at most $O(W|E|)$ since each path is of length $O(W)$. Thus, we have established that the size of the lifted Markov chain is at most $O(W|E|) = O^*(|E|/\Phi(P))$. \square

D. Useful Claims

We state and prove two useful claims which plays a key role in proving Lemma 11.

Claim 13: Let P_1, P_2 be reversible Markov chains with their stationary distributions π_1, π_2 , respectively. If there exist positive constants α, β, c, d such that $P_1 \geq \alpha P_2$, $P_1 \geq \beta I$, and $c\pi_2 \leq \pi_1 \leq d\pi_2$, then

$$\lambda_{P_1} \geq \min\left(\frac{\alpha c}{d^2} \lambda_{P_2}, 2\beta\right).$$

Proof: From the min-max characterization of the spectral gap (see, e.g., [15, p. 176]) for the reversible Markov chain, it follows that

$$\begin{aligned} \lambda_{P_1} &= \inf_{\psi: V \rightarrow \mathbb{R}} \left(\frac{\sum_{i,j \in V} (\psi(i) - \psi(j))^2 (\pi_1)_i (P_1)_{ij}}{\sum_{i,j \in V} (\psi(i) - \psi(j))^2 (\pi_1)_i (\pi_1)_j} \right) \\ &\geq \left(\frac{\alpha c}{d^2}\right) \inf_{\psi: V \rightarrow \mathbb{R}} \left(\frac{\sum_{i,j \in V} (\psi(i) - \psi(j))^2 (\pi_2)_i (P_2)_{ij}}{\sum_{i,j \in V} (\psi(i) - \psi(j))^2 (\pi_2)_i (\pi_2)_j} \right) \\ &= \left(\frac{\alpha c}{d^2}\right) \lambda_{P_2}. \end{aligned}$$

¹¹The precise bound is $O(|E|W)$.

The smallest eigenvalue of P_1 is greater than $2\beta - 1$ because $P_1 \geq \beta I$. So, the distance between the smallest eigenvalue and -1 is greater than 2β . This completes the proof. \square

Claim 14: Let P_1, P_2 be Markov chains with their stationary distributions π_1, π_2 , respectively. Now, suppose P_2 is reversible. (P_1 is not necessarily reversible.) If there exist positive constants α, β, c, d such that $P_1 \geq \alpha P_2$, $P_1 \geq \beta I$ and $c\pi_2 \leq \pi_1 \leq d\pi_2$, then

$$\lambda_{P_1 P_1^*} \geq \min\left(\frac{\alpha \beta c}{d^2} \lambda_{P_2}, 2\beta^2\right).$$

Proof: $P_1 P_1^*$ is a reversible Markov chain which has π_1 as its stationary distribution. Because $P_1^* \geq \beta I$, $P_1 P_1^* \geq \alpha P_2 P_1^* \geq \alpha \beta P_2$. Also, $P_1 P_1^* \geq \beta^2 I$. Now, the proof follows from Claim 13. \square

VII. CONCLUSION

Motivated by applications arising in emerging networks such as sensor networks, peer-to-peer networks, and surveillance network of unmanned vehicles, we consider the question of designing fast linear iterative algorithms for computing the average of numbers in a network. We presented a novel construction of such an algorithm by designing the fastest mixing non-reversible Markov chain on any given graph. Our Markov chain obtained through a new notion denoted by pseudo-lifting. We apply our constructions to graphs with geometry, or graphs with doubling dimension. By using their topological properties explicitly, we obtain fast and slim pseudo-lifted Markov chains. The effectiveness (and optimality) of our constructions are explained through various examples. As a byproduct, our result provides the fastest mixing Markov chain for any given graph which should be of interest in its own right. Our results should naturally find their applications in the context of distributed optimization, estimation, and control.

We note that the pseudo-lifting presented here is based on a two-level ‘‘hierarchical star’’ topology. This construction is less robust to node failures. For example, failure of ‘‘root’’ node can increase the mixing time drastically. To address this, one may alternatively use a ‘‘hierarchical expander’’ based pseudo-lifting. That is, in place of the ‘‘star’’ topology in the pseudo-lifting, utilize the ‘‘expander’’ topology. This will naturally make the construction more robust without loss of performance. Of course, this will complicate the mixing time analysis drastically. This is where our method developed in the expander-based lifting will be readily useful.

REFERENCES

- [1] D. J. Aldous, ‘‘Some inequalities for reversible Markov chains,’’ *J. London Math. Soc.*, vol. 25, pp. 564–576, 1982.
- [2] P. Assouad, ‘‘Plongements lipschitziens dans \mathbb{R}^n ,’’ *Bull. Soc. Math. France*, vol. 111, no. 4, pp. 429–448, 1983.
- [3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, ‘‘Convergence in multiagent coordination, consensus, and flocking,’’ in *Proc. Joint 44th IEEE Conf. Decision and Control and European Control Conference (CDC-ECC’05)*, Seville, Spain, Dec. 2005, pp. 2996–3000.
- [4] S. Boyd, P. Diaconis, J. Sun, and L. Xiao, ‘‘Fastest mixing Markov chain on a path,’’ *Amer. Math. Monthly*, vol. 113, no. 1, pp. 70–74, 2006.

- [5] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, 2004.
- [6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. IEEE INFOCOM 2005*, Miami, FL, 2005, pp. 1653–1664.
- [7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2508–2530, 2006.
- [8] F. Chen, L. Lovász, and I. Pak, "Lifting Markov chains to speed up mixing," in *Proce. 36th Annu. ACM Symp. Theory of Computing (STOC'99)*, New York, 1999, pp. 275–281, published by ACM Press.
- [9] S. Dasgupta and Y. Freund, "Random projection trees and low dimensional manifolds," in *Proc. 40th Annu. ACM Symp. Theory of Computing (STOC'08)*, New York, 2008, pp. 537–546, published by ACM Press.
- [10] P. Diaconis, S. Holmes, and R. Neal, "Analysis of a non-reversible Markov chain sampler," *Ann. Appl. Probab.*, vol. 10, pp. 726–752, 2000.
- [11] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient aggregation for sensor networks," in *Proc. 5th Int. ACM/IEEE Symp. Information Processing in Sensor Networks (IPSN'06)*, Nashville, TN, Apr. 2006, pp. 69–76.
- [12] A. Gupta, R. Krauthgamer, and J. R. Lee, "Bounded geometries, fractals, and low-distortion embeddings," in *Proc. 44th Annu. IEEE Symp. Foundations of Computer Science (FOCS'03)*, Washington, DC, 2003, p. 534, published by the IEEE Computer Society.
- [13] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.
- [14] J. Heinonen, Ed., *Lectures on Analysis on Metric Spaces*. Berlin, Germany: Springer, 2001.
- [15] R. Horn and C. Johnson, Eds., *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press., 1985.
- [16] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Aut. Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [17] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annu. IEEE Symp. Foundations of Computer Science (FOCS'03)*, Washington, DC, 2003, p. 482, published by the IEEE Computer Society.
- [18] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. 36th Annu. ACM Symp. Theory of Computing (STOC'04)*, New York, 2004, pp. 561–568, published by ACM Press.
- [19] C. Kenyon, E. Mossel, and Y. Peres, "Glauber dynamics on trees and hyperbolic graphs," in *Proc. IEEE Symp. Foundations of Computer Science*, Las Vegas, NV, 2001, pp. 568–578.
- [20] P. Kolman and C. Scheideler, "Improved bounds for the unsplitable flow problem," in *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA'02)*, Philadelphia, PA, 2002, pp. 184–193, published by SIAM.
- [21] M. Koubarakis, C. Tryfonopoulos, S. Idreos, and Y. Drougas, "Selective information dissemination in p2p networks: Problems and solutions," *SIGMOD Rec.*, vol. 32, no. 3, pp. 71–76, 2003.
- [22] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," in *Proc. 29th Annu. Symp. Foundations of Computer Science (FOCS'88)*, White Plains, NY, 1988, pp. 422–431.
- [23] L. Lovasz and P. Winkler, D. Aldous and J. Propp, Eds., "Mixing times," *Microsurveys in Discrete Probability*, ser. DIMACS Series in Discrete Mathematics and Theory of Computer Science, pp. 85–133, 1998.
- [24] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: A tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.
- [25] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.
- [26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.
- [27] D. Mosk-Aoyama and D. Shah, "Fast distributed algorithms for computing separable functions," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 2997–3007, Jul. 2008.
- [28] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, LIDS report 2755, to be published.
- [29] P. Diaconis and L. Saloff-Coste, "Moderate growth and random walk on finite groups," *Geometric and Functional Analysis*, vol. 4, no. 1, pp. 1–36, 1994.
- [30] K. Savlja, F. Bullo, and E. Frazzoli, "On traveling salesperson problems for Dubins' vehicle: Stochastic and dynamic environments," in *Proc. IEEE Conf. Decision and Control and European Control Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 4530–4535.
- [31] J. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, Lab. Information and Decision Systems, MIT, Cambridge, MA, 1984.
- [32] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Contr. Lett.*, vol. 53, pp. 65–78, 2004.

Kyomin Jung received the B.Sc. degree from Seoul National University, Seoul, Korea, in 2003 and the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 2009

He is an Assistant Professor with the Department of Electrical Engineering and Computer Science at KAIST. During his Ph.D. study he worked at Microsoft Research, Cambridge (2008), IBM T. J. Watson Research Center, Yorktown Heights, NY (2007), Bell Labs, Murray Hill, NJ (2006), and Samsung Advanced Institute of Technology (2005), respectively, as a research summer intern. His main research interest includes routing/scheduling algorithms and performance analysis in networks, and optimization problems in statistical inference.

Dr. Jung won a gold medal at the IMO (International Mathematical Olympiad) in 1995.

Devavrat Shah received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1999 and the Ph.D. degree from Stanford University, Stanford, CA, in 2004.

He is currently a Jamieson Career Development Associate Professor with the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT), Cambridge. His research interests include statistical inference and network algorithms.

Dr. Shah was the corecipient of the IEEE INFOCOM best paper award in 2004 and ACM SIGMETRICS/Performance best paper award in 2006. He received 2005 George B. Dantzig best dissertation award from the INFORMS. He is the recipient of the first ACM SIGMETRICS Rising Star Award 2008 for his work on network scheduling algorithms.

Jinwoo Shin received the B.Sc. degree from Seoul National University, Seoul, Korea, with double majors, mathematics and computer science.

He is working towards the Ph.D. degree in the Department of Mathematics at the Massachusetts Institute of Technology (MIT), Cambridge. Before joining the doctoral program at MIT, he completed military duty of South Korea via working at Penta Security Inc. for 3.5 years as a programmer of computer-security software. His primary research interests are theoretical aspects of communication networks.

Mr. Shin won a silver medal at the IMO (International Mathematical Olympiad) in 1995. He was a corecipient of ACM SIGMETRICS/Performance Kenneth C. Sevcik outstanding student paper award in 2009.