Decentralized Computation of Effective Resistances and Acceleration of Distributed Optimization Algorithms

Bugra Can, *Student Member, IEEE* , Saeed Soori, *Student Member, IEEE,* Necdet Serhat Aybat, *Member, IEEE,* Maryam Mehri Dehnavi, *Member, IEEE,* Mert Gürbüzbalaban, *Member, IEEE*

***Abstract*—The effective resistance between a pair of nodes in a weighted undirected graph is defined as the potential difference induced between them when a unit current is injected at the first node and extracted at the second node, treating edge weights as the conductance values of edges. The effective resistance is a key quantity of interest in many applications and fields including solving linear systems, Markov Chains and continuous-time averaging networks. We develop an efficient linearly convergent distributed algorithm for computing effective resistances and demonstrate its performance through numerical studies. We also apply our algorithm to the consensus problem where the aim is to compute the average of node values in a distributed manner based on weighted averaging with immediate neighbors. We show that the distributed algorithm we developed for effective resistances can be used to design a weight matrix which can help pass the information among neighbors more *effectively*, accelerating the convergence of the classical consensus iterations considerably by a factor depending on the network structure. We also present an application of our effective resistance-based framework to accelerate distributed optimization algorithms including the EXTRA and DPGA-W.**

***Index Terms*—Effective resistance, graph, distributed optimization, consensus, Laplacian matrix, Kaczmarz method**

I. Introduction

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E}, w)$ be an undirected, weighted and connected graph defined by the set of nodes (agents) $\mathcal{N} = \{1, \dots, n\}$, the set of edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$, and the edge weights $w_{ij} > 0$ for $(i,j) \in \mathcal{E}$. Since $\mathcal{G}$ is undirected, we assume that both (i,j) and (j,i) refer to the same edge when it exists, and for all $(i,j) \in \mathcal{E}$, we set $w_{ji} = w_{ij}$. Identifying the weighted graph $\mathcal{G}$ as an electrical network in which each edge (i,j) corresponds to a branch of conductance w_{ij}, the effective resistance R_{ij} between a pair of nodes i and j is defined as the voltage potential difference induced between them when a unit current is injected at i and extracted at j.

The effective resistance, also known as the resistance distance, is a key quantity of interest to compute in many applications and algorithmic questions over graphs. It defines a metric on graphs providing bounds on its conductance [2], [3]. Furthermore, it is closely associated with the hitting time and commute time for a random walk[1] on the graph G such that the probability of a transition from i to $j \in \mathcal{N}_i$ is $w_{ij} / \sum_{j' \in \mathcal{N}_i} w_{ij'}$ where $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : \ w_{ij} > 0\}$ denotes the set of neighboring nodes of $i \in \mathcal{N}$; therefore, it arises naturally for studying random walks over graphs and their mixing time properties [4], [5], [6], continuous-time averaging networks including consensus problems in distributed optimization [4]. Other prominent applications include distributed control and estimation [7], solving symmetric diagonally dominant (SDD) linear systems [8], deriving complexity bounds in the Asymmetric Traveling Salesman Problem [9], design and control of communication networks [10], [11], spectral sparsification of graphs [12] and collaborative recommendation systems [13]. There exist *centralized* algorithms for computing or approximating effective resistances accurately which require global communication beyond local communication among the neighboring agents [8], [14]. They are based on computing or approximating the entries of the pseudoinverse $\mathcal{L}^\dagger$ of the Laplacian matrix $\mathcal{L}$, based on the identity $R_{ij} = \mathcal{L}^\dagger_{ii} + \mathcal{L}^\dagger_{jj} - 2\mathcal{L}^\dagger_{ij}$ [8]. However, such centralized algorithms are *impractical* or *infeasible* for several key applications in multi-agent systems where only local communications between the neighboring agents are allowed (see e.g. [15], [16], [17]); this motivates the development of *distributed algorithms* for computing effective resistances which only relies on the information exchange among immediate neighbors. In these applications, communication among the agents is typically the bottleneck rather than local computations by the agents; and this requires careful development of distributed algorithms that are efficient in terms of total number of communications required. Prominent examples include, least square and more general regression or estimation problems over graphs and multi-agent networks [18], [19], [20], [21], [22], formation control of moving agents with noisy measurements and stability of multi-vehicle swarms [7], clustering and sparsification of multi-agent networks [23], [8].

Our first attempt for computing effective resistances in a decentralized way appeared in a short conference paper [1].

A short conference version of this work appeared in [1] at the IEEE GlobalSIP'17 conference.

Bugra Can and M. Gürbüzbalaban are with the Department of Management and Information Sciences, Rutgers Business School, Piscataway, NJ-08854, USA.(email: bugra.can@rutgers.edu; mg1366@rutgers.edu)

Saeed Soori is with Department of Electrical & Computer Engineering Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA. (email: saeed.soori@rutgers.edu)

Necdet Serhat Aybat is with Industrial and Manufacturing Engineering Department Penn State University, University Park, PA 16802-4400, USA. (email: nsa10@psu.edu)

Maryam Mehri Dehnavi is with Department of Computer Sciences, University of Toronto, Toronto, Canada.(email: mmehride@cs.toronto.edu)

[1] The hitting time is the expected number of steps of a random walk starting from i until it first visits j. The commute time C_{ij} is the expected number of steps required to go from i to j and from j to i back again.

Compared to the short conference version [1], in this longer journal paper we provide some new important theoretical and empirical results. First, we provide a theory to support our empirical findings in [1]. More precisely, we now show that the iterate sequence generated by the normalized randomized Kaczmarz proposed in [1] for computing effective resistances in a distributed manner linearly converges with a rate at most as large as the standard randomized Kaczmarz (RK), i.e., with a better rate (see Proposition 1). Moreover, we now provide a theoretical analysis of the effective resistance-based randomized gossiping for consensus which was missing in [1]. Briefly, we bound the time required to compute an inexact average using graph conductance-based and spectral-based analyses and compare these bounds corresponding to effective resistance-based and classic randomized gossiping methods. We were able to show that averaging time with resistance weights is $\Theta(n)$ faster than that of classic gossiping on a barbell graph. Furthermore, we prove that for some graphs, the averaging time with resistance weights can be faster than known performance bounds for the averaging time with Metropolis weights. Second, we added numerical experiments that show that our framework can accelerate DPGA-W and EXTRA algorithms for consensus optimization.

Contributions. To our knowledge, there has been no systematic study of (linearly convergent) efficient *distributed* algorithms for computing effective resistances. In this work, first we discuss how existing algorithms in the distributed optimization literature for solving linear systems can be adapted to solve this problem. After showing that a naive implementation of existing distributed optimization algorithms, e.g., the EXTRA algorithm [24], is inefficient in terms of the convergence and communication requirements, we focus on Kaczmarz methods [8]. We propose a distributed variant of the Kaczmarz method to compute the effective resistances and show that this algorithm is linearly convergent while being efficient in terms of total number of local communications carried out. In particular, numerical experiments suggest finite convergence of our algorithms which is of independent interest. Then, we apply our results to the consensus problem which is the problem of computing the average of node values over a network in a distributed manner [16]. We show that classic gossiping algorithms can be accelerated using local information about effective resistances. Specifically, we propose a variant of the classical *asynchronous* consensus protocol and show that we can accelerate the convergence considerably by a factor depending on the underlying network. The main idea is to use the distributed algorithm we developed for effective resistances to design a weight matrix which can help pass the information among neighbors more *effectively*. Second, we consider the consensus optimization problem, where the agents connected on a network aim to collaboratively solve the optimization problem $\min_{x\in\mathbb{R}^p} f(x) := \sum_{i=1}^n f_i(x)$ where $f_i(x) : \mathbb{R}^p \to \mathbb{R}$ is a cost function only available to (node) agent i. This problem includes a number of key problems in supervised learning including distributed regression and logistic regression or more generally distributed empirical risk minimization problems [25], [26]. The consensus iterations are a building block of many existing state-of-the-art distributed consensus optimization algorithms such as the EXTRA algorithm and the distributed proximal gradient algorithm (DPGA-W) [27] for consensus optimization. We show through numerical experiments that our framework based on effective resistances can accelerate the EXTRA and DPGA-W algorithms for consensus optimization. We believe our method and framework have far-reaching potential for accelerating many other distributed algorithms including distributed subgradient and ADMM methods, and this will be the subject of future work.

Related work. For consensus problems, there are some alternative methods to accelerate the classical consensus protocols. The approach in [28] is a synchronous algorithm and builds on modifying the weights depending on the degree of the neighbors based on Metropolis weights and a momentum averaging scheme. There are also other approaches based on momentum averaging [29], [30], [31], min-sum splitting [32] and Chebyshev acceleration [33], [34] to accelerate the speed of the consensus methods. Our approach is orthogonal to these alternative approaches and it aims at improving the communication efficiency of a consensus optimization algorithm and can be used in principle on top of such acceleration schemes for consensus iterations from the literature where randomized sampling of the nodes and weighted averaging (of local variables at nodes) are used. We also note that since effective resistance R_{ij} is proportional to the commute time C_{ij} between nodes i and j, one could in principle generate the sample paths of T random walks between these nodes to estimate C_{ij} up to accuracy $\mathcal{O}(1/\sqrt{T})$ in a distributed fashion with a Monte Carlo approach [35], [13]. However, this technique suffers from slow (sublinear) convergence and does not scale well to large graphs – which is typical for the random walk-based Monte Carlo methods, in contrast with the linearly convergent algorithms we are providing in this work that are efficient in terms of total number of communications among the nodes.

Outline. In Section II, we introduce our decentralized algorithm for computing effective resistances. In Section III, we propose an effective resistance-based asynchronous gossiping algorithm for solving the consensus problem and show its linear convergence. In Section IV, we provide a number of theoretical convergence guarantees for this algorithm that illustrates the performance improvement that can be obtained with our approach. In Section V, we provide numerical results for algorithms given in Sections II and III and illustrate that our framework based on effective resistances can accelerate EXTRA and DPGA-W algorithms for consensus optimization. Finally, in Section VI, we give a summary of our results and discuss future potential work.

Notation. Let $|S|$ denote the cardinality of a set S and $\lfloor . \rfloor$ denote the floor function. We define $d_i \triangleq |\mathcal{N}_i|$ as the degree of $i \in \mathcal{N}$, and $m \triangleq |\mathcal{E}|$. Throughout the paper, $\mathcal{L} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ denotes the weighted Laplacian of $\mathcal{G}$, i.e., $Ł_{ii} = \sum_{j\in\mathcal{N}_i} w_{ij}$, $\mathcal{L}_{ij} = -w_{ij}$ if $j \in \mathcal{N}_i$, and equals to 0 otherwise. The set $\mathbb{S}^n$ denotes the set of $n \times n$ real symmetric matrices. For an $n \times m$ matrix A, $\mathbf{Null}(A) \triangleq \{x \in \mathbb{R}^m \mid Ax = 0\}$ and $\mathbf{Rank}(A) \triangleq \{y \in \mathbb{R}^n \mid \exists x \in \mathbb{R}^m \text{ s.t. } y = Ax\}$. We use the notation $Z = [z_i]_{i=1}^n$ where z_i's are either the columns or rows of the matrix Z depending on the context. $\mathbf{1}$ is the column vector with all entries equal to 1, and $\mathbf{I}$ is the identity matrix. We let $||x||_p$ denote the L_p norm of a vector x for $p \geq 1$. We also let $\|A\|_F$ to denote the Frobenius norm of a matrix A. A square matrix A is *doubly stochastic* if all of its entries are non-negative and all its rows and columns sum up to 1. We say that a square matrix A is *weakly diagonally dominant* if its diagonal entries A_{ii} satisfy the inequality $|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|$ for every i. Let f and g be real-valued functions defined over positive integers. We say $f(n) = \mathcal{O}(g(n))$ if f is bounded above by g asymptotically, i.e. there exist constants $k_1 > 0$ and n_0 such that $f(n) \leq k_1 \cdot g(n)$ for all $n > n_0$. We say $f(n) = \Theta(g(n))$ if f is bounded both above and below by g asymptotically, i.e., there exist constants $k_1, k_2 > 0$ and n_0 such that $k_2 \cdot g(n) \leq f(n) \leq k_1 \cdot g(n)$ for all $n > n_0$. Similarly, we say $f(n) = \Omega(g(n))$ if there exist constants $c > 0$ and n_0 such that $f(n) \geq c\, g(n)$ for every $n > n_0$. Finally, let $\log(x)$ denote the natural logarithm of x (logarithm having the mathematical constant e as a base).

II. Methodology

It is well known that the Laplacian matrix $\mathcal{L}$ of an undirected graph is symmetric and positive semidefinite; and since $\mathcal{G}$ is connected, the nullspace of $\mathcal{L}$ is spanned by $\mathbf{1}$. In particular, consider the eigenvalue decomposition $\mathcal{L} = \sum_{i=1}^{n} \lambda_i u_i u_i^\top$; we have $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ and $u_1 = \frac{1}{\sqrt{n}}\mathbf{1}$. Recall that we would like to compute $\mathcal{L}^\dagger = \sum_{i=2}^{n} \frac{1}{\lambda_i} u_i u_i^\top$ in a decentralized way. First, we are going to describe a naive way to solve this problem which would converge with a linear rate, but require storing and communicating $n \times n$ matrices among the neighboring nodes. Next, we discuss that $\mathcal{L}^\dagger$ can be computed in a distributed way using the (randomized) Kaczmarz (RK) method with significantly less communication burden.

A. A consensus-based naive method for computing $\mathcal{L}^\dagger$:

Let $\theta \geq \lambda_2$ and define $\bar{\mathcal{L}} \triangleq \mathcal{L} + \frac{\theta}{n}\mathbf{1}\mathbf{1}^\top$, i.e., $\bar{\mathcal{L}} = \theta u_1 u_1^\top + \sum_{i=2}^{n} \lambda_i u_i u_i^\top$; hence, $\bar{\mathcal{L}}^{-1} = \mathcal{L}^\dagger + \frac{1}{\theta} u_1 u_1^\top$. To compute $\bar{\mathcal{L}}^{-1}$, consider solving $(P) : \min_{X \in \mathbb{S}^n} f(X) \triangleq \frac{1}{2}\left\|\bar{\mathcal{L}}X - \mathbf{I}\right\|_F^2$. Note that f is strongly convex with modulus λ_2 since $\theta \geq \lambda_2$; moreover, such θ can be chosen easily in certain cases. For instance, for unweighted $\mathcal{G}$, i.e., $w_{ij} = 1$ for $(i,j) \in \mathcal{E}$, it is known that $\lambda_2 \leq \min_{i \in \mathcal{N}} d_i$; hence, θ could be chosen after running a min-consensus algorithm over $\mathcal{G}$. To solve (P) in a decentralized manner, we will exploit connectivity of $\mathcal{G}$. Let $\bar{\ell}_i \in \mathbb{R}^n$ be a column vector for $i \in \mathcal{N}$ such that $\bar{\mathcal{L}} = [(\bar{\ell}_i)^\top]_{i \in \mathcal{N}}$, i.e., $(\bar{\ell}_i)^\top$ denotes the i-th row of $\bar{\mathcal{L}}$. (P) can be equivalently written as follows:

$$(P') : \min_{X_i \in \mathbb{S}^n,\ i \in \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \left\|X_i \bar{\ell}_i - e_i\right\|_2^2 :\ X_i = X_j \ \forall\ (i,j) \in \mathcal{E} \right\},$$

where e_i denotes the i-th standard basis vector of $\mathbb{R}^n$. Although this problem is not strongly convex in $[X_i]_{i \in \mathcal{N}}$, there is a way to regularize the objective $\bar{f}([X_i]_{i \in \mathcal{N}}) \triangleq \sum_{i \in \mathcal{N}} \left\|X_i \bar{\ell}_i - e_i\right\|_2^2$ to make it strongly convex. Indeed, it can be shown that for $\alpha > 0$ sufficiently large, $\bar{f}_\alpha \triangleq \bar{f} + \alpha r$ is strongly convex in $[X_i]_{i \in \mathcal{N}}$, where $r([X_i]_{i \in \mathcal{N}}) \triangleq \sum_{(i,j) \in \mathcal{E}} \|X_i - X_j\|_F^2$; and one can equivalently consider $\min\{\bar{f}_\alpha([X_i]_{i \in \mathcal{N}}) :\ X_i = X_j\,(i,j) \in \mathcal{E}\}$ – for details, see [36], [37]. In particular, the algorithm EXTRA in [24] exploits a similar restricted strong convexity argument and achieves a linear convergence rate for the iterate sequence. That being said, the communication overhead is the main problem with this approach for solving (P'). In fact, at each iteration k, each node $i \in \mathcal{N}$ communicates its local estimate X_i^k to its neighbors in $\mathcal{N}_i$; thus, each iteration of these consensus based methods would require $\mathcal{O}(2|\mathcal{E}|n^2)$ real variable communications in total, e.g., EXTRA. Next, we discuss the distributed implementation of the RK method to compute $\mathcal{L}^\dagger$, which would prove itself as a more communication efficient and practical method.

B. Distributed Kaczmarz method for computing $\mathcal{L}^\dagger$:

Consider a *consistent* linear system $Ax = b$, where $A = [a_i^\top]_{i=1}^m \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Suppose A has no rows with all zeros, and let $x^* = \operatorname{argmin}\{\|x\|_2 :\ Ax = b\}$. In [38], it is shown that x^* can be computed using a randomized Kaczmarz method. In particular, it follows from the results in [38] that starting from $x^0 \in \mathbf{Null}(A)$, the method displayed in Algorithm 1 produces $\{x^k\}_{k \geq 1}$ such that $\mathbb{E}[\left\|x^k - x^*\right\|_2^2] \leq \rho^k \left\|x^0 - x^*\right\|_2^2$ for $k \geq 0$ with $\rho \triangleq 1 - \lambda_{\min}^+(A^\top H A)$ where $\lambda_{\min}^+(\cdot)$ denotes the smallest positive eigenvalue and $H = \sum_{i=1}^m p_i \frac{1}{\|a_i\|_2^2} e_i e_i^\top$; furthermore, $1 - \frac{1}{\mathbf{Rank}(A)} \leq \rho < 1$. Note that fixing $p_i = \|a_i\|_2^2 / \|A\|_F^2$ gives us the randomized Kaczmarz in [39], [40].

Algorithm 1: RK($\{p_i\}_{i=1}^m$) – Randomized Kaczmarz

```
Initialization: $x^0 \in \mathbf{Null}(A)$
for $k \geq 0$ do
    Pick $i \in \{1, \ldots, m\}$ with probability (w.p) $p_i$
    $x^{k+1} \leftarrow x^k - \frac{1}{\|a_i\|^2}(a_i^\top x^k - b_i)a_i$
```

Note $\mathcal{L}\mathcal{L}^\dagger = \sum_{i=2}^n u_i u_i^\top$ and $\mathbf{I} = \sum_{i=1}^n u_i u_i^\top$; hence, $\mathcal{L}\mathcal{L}^\dagger = \mathbf{I} - u_1 u_1^\top = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Although the solution set $\{X \in \mathbb{S}^n : \mathcal{L}X = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\}$ has infinitely many elements, it is well-known that $\mathcal{L}^\dagger$ is the unique solution to

$$\mathcal{L}^\dagger = \operatorname*{argmin}_{X \in \mathbb{S}^n}\{\|X\|_F :\ \mathcal{L}X = B\}, \tag{1}$$

where $B \triangleq \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Let $x^l, b^l \in \mathbb{R}^n$ for $l \in \mathcal{N}$ be column vectors such that $X = [x^l]_{l \in \mathcal{N}}$ and $B = [b^l]_{l \in \mathcal{N}}$, i.e., $b^l = e_l - \frac{1}{n}\mathbf{1}$. Note n columns of $\mathcal{L}^\dagger$ can be computed in parallel:

$$x_*^l \triangleq \operatorname*{argmin}_{x \in \mathbb{R}^n}\{\|x\|_2 :\ \mathcal{L}x = b^l\}, \quad l \in \mathcal{N}, \tag{2}$$

i.e., $\mathcal{L}^\dagger = [x_*^l]_{l \in \mathcal{N}}$. Since $\mathcal{L}^\dagger \mathbf{1} = \mathbf{0}$, $x_*^n = -\sum_{l=1}^{n-1} x_*^l$. Thus, one does not need to solve for all $l \in \mathcal{N}$; it suffices to compute $\{x_*^l\}_{l \in \mathcal{N} \setminus \{n\}}$ and calculate x_*^n from these.

Let $\{x^{l,k}\}_{k \geq 1}$ be the sequence generated when RK implemented on (2) for $l \in \{1, \ldots, n-1\}$. In Algorithm 2, we summarized the distributed nature of RK steps assuming that each $i \in \mathcal{N}$ has an exponential clock with rate $r_i > 0$, and when its clock ticks, the node i wakes up and communicates with its neighbors $j \in \mathcal{N}_i$ on $\mathcal{G}$. More precisely, consider the resulting superposition of these point processes, and let $\{t_k\}_{k \in \mathbb{Z}_+}$ be the times such that one of the clocks ticks; hence, for all $k \geq 0$, the node that wakes up at time t_k is node i with probability $p_i = r_i / \sum_{j\in\mathcal{N}} r_i$, i.e., $\{t_k\}_{k\geq 0}$ denotes the arrival times of a Poisson process with rate $\sum_{j\in\mathcal{N}} r_j$.

Algorithm 2: D-RK($\{r_i\}_{i\in\mathcal{N}}$) – Decentralized RK

```
Initialization: $x_i^{l,0} \leftarrow 0$ for $l \in \mathcal{N} \setminus \{n\}$ and $i \in \mathcal{N}$
for $k \geq 0$ do
    At time $t_k$, $i \in \mathcal{N}$ wakes up w.p. $p_i = \frac{r_i}{\sum_{j\in\mathcal{N}} r_i}$
    for $l \in \{1, \ldots, n-1\}$ do
        Node $i$ requests and receives $x_j^{l,k}$ from $j \in \mathcal{N}_i$
        Node $i$ computes and sends $q_i^{l,k}$ to all $j \in \mathcal{N}_i$
          $q_i^{l,k} = \frac{1}{\sum_{j\in\mathcal{N}_i\cup\{i\}} \mathcal{L}_{ij}^2}(\sum_{j\in\mathcal{N}_i\cup\{i\}} \mathcal{L}_{ij} x_j^{l,k} - b_i^l)$
        Each $j \in \mathcal{N}_i \cup \{i\}$ updates $x_j^{l,k+1} \leftarrow x_j^{l,k} - \mathcal{L}_{ij} q_i^{l,k}$
```

For $k \geq 0$, let $X^k \triangleq [x^{l,k}]_{l\in\mathcal{N}}$ be the concatenation of D-RK sequence, where $x^{n,k} \triangleq -\sum_{l=1}^{n-1} x^{l,k}$, and define

$$S = \mathbf{diag}(s) \quad \text{where} \quad s_i \triangleq \sum_{j\in\mathcal{N}_i\cup\{i\}} \mathcal{L}_{ij}^2 \text{ for } i \in \mathcal{N}. \tag{3}$$

According to [38], [39], for $r_i = s_i$, we get $H = \frac{1}{\|\mathcal{L}\|_F^2}\mathbf{I}$, and this implies linear convergence of $\{X^k\}_{k\geq 0}$ to $\mathcal{L}^\dagger$ with rate

$$\rho \triangleq 1 - \left(\frac{\lambda_{\min}^+(\mathcal{L})}{\|\mathcal{L}\|_F}\right)^2, \tag{4}$$

i.e., $\mathbb{E}[\|X^k - \mathcal{L}^\dagger\|_F^2] \leq \rho^k \|\mathcal{L}^\dagger\|_F^2$ for $k \geq 0$. Moreover, for each $i \in \mathcal{N}$, when node i wakes up, Decentralized Randomized Kaczmarz (D-RK) defined by Algorithm 2 requires $2d_i(n-1)$ communications – each communication i sends/receives a real variable to/from a neighboring node in $\mathcal{N}_i$; hence, at each iteration, i.e., at each time a node wakes up, the expected number of communication per iteration is $N = \sum_{i\in\mathcal{N}} 2p_i d_i(n-1) \leq 2d_{\max}(n-1)$. In particular, for unweighted graphs, i.e., $w_{ij} = 1$ for $(i,j) \in \mathcal{E}$, we have $p_i = \frac{d_i(d_i+1)}{2m+\sum_{j\in\mathcal{N}} d_j^2}$ for $i \in \mathcal{N}$.

Instead of (1), consider implementing D-RK on a normalized system $S^{-1/2}\mathcal{L}X = S^{-1/2}B$ to obtain better convergence rate in practice – i-th equation in this normalized system can be computed locally at $i \in \mathcal{N}$. For this system, where all the rows have unit norm, one can set $r_i = r$ for some $r > 0$ for all $i \in \mathcal{N}$ – hence, nodes wake up with uniform probability, i.e., $p_i = \frac{1}{n}$ for $i \in \mathcal{N}$; for this choice of equal clock rates, $H = \frac{1}{n}\mathbf{I}$ and $\{X^k\}_k$ converges linearly to $\mathcal{L}^\dagger$ with rate

$$\rho_S \triangleq 1 - \frac{1}{n}\lambda_{\min}^+(\mathcal{L}S^{-1}\mathcal{L}). \tag{5}$$

Moreover, the expected number of communication per iteration is $N = 4m\frac{n-1}{n} \leq 4m$. In all experiments on small world random networks – see the definition in the numerical section – D-RK implemented on the normalized system worked significantly better than directly implementing it on (1) (see Fig. 4). The next proposition formalizes this numerical observation and shows that the normalized system has a faster convergence rate. The proof is deferred to the Appendix A.

Proposition 1. *It holds that*

$$\frac{1}{n}\lambda_{\min}^+(\mathcal{L}S^{-1}\mathcal{L}) \geq \left(\frac{\lambda_{\min}^+(\mathcal{L})}{\|\mathcal{L}\|_F}\right)^2, \tag{6}$$

where S is given by (3)*. Then, it follows that $\rho_S \leq \rho$ where ρ and ρ_S are defined by* (4) *and* (5)*.*

In the next section, we discuss how effective resistances can be used to accelerate classical gossiping algorithms for solving the consensus problem.

III. Effective resistance-based Consensus

Let $y^0 \in \mathbb{R}^n$ be a vector such that the i-th component represents the initial value at node i, and let $\bar{y} \triangleq \sum_{i=1}^n y_i^0/n$ be the average. In consensus algorithms, the aim is to compute $\bar{y}$ at each node in a distributed manner. As in Section II-B, we assume that each $i \in \mathcal{N}$ has an exponential clock with rate $r_i > 0$. If a node wakes up at time t_k, it is node i with probability (w.p) p_i. Given that the clock of node i ticks at time t_k and the node i wakes up, the conditional probability that it picks *one* of its neighbors $j \in \mathcal{N}_i$ is given by $p_{j|i} \in (0,1)$, where $\sum_{j\in\mathcal{N}_i} p_{j|i} = 1$. Next, nodes i and j exchange their local variables y_i^k and y_j^k. We assume that each node $i \in \mathcal{N}$ knows $\{R_{ij}\}_{j\in\mathcal{N}}$, where $R_{ij} = \mathcal{L}_{ii}^\dagger + \mathcal{L}_{jj}^\dagger - 2\mathcal{L}_{ij}^\dagger$ for $(i,j) \in \mathcal{E}$.

Algorithm 3: Randomized Gossiping

```
Initialization: $y^0 = [y_1^0, y_2^0, \ldots, y_n^0]^\top \in \mathbb{R}^n$
for $k \geq 0$ do
    At time $t_k$, $i \in \mathcal{N}$ wakes up w.p. $p_i = r_i / \sum_{j\in\mathcal{N}} r_j$
    Picks $j \in \mathcal{N}_i$ randomly w.p. $p_{j|i}$
    $y_i^{k+1} \leftarrow \frac{y_i^k + y_j^k}{2}$,   $y_j^{k+1} \leftarrow \frac{y_i^k + y_j^k}{2}$
```

Assuming that *there are no self-loops*, for each $i \in \mathcal{N}$, let

$$P_{ii} = 0; \quad P_{ij} := p_i p_{j|i}, \quad \forall\, j \in \mathcal{N}_i, \tag{7a}$$

$$P_{ij} := 0, \quad \forall j \in \mathcal{N} \setminus \mathcal{N}_i, \tag{7b}$$

where P_{ij} is the (unconditional) probability that the edge (i,j) is activated by the node i. Note that by definition, we have $\sum_{ij} P_{ij} \triangleq \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} P_{ij} = 1$. We will be considering two different consensus protocols, where in both protocols nodes operate as in Algorithm 3 but with different $\{p_i\}_{i\in\mathcal{N}}$ and $\{p_{j|i}\}_{j\in\mathcal{N}_i}$ for $i \in \mathcal{N}$.

1) Classic Randomized Gossiping: At each iteration k, each edge $(i,j) \in \mathcal{E}$ has equal probability of being activated. If an edge (i,j) is activated at iteration k, the nodes take average of their local variables y_i^k and y_j^k. This algorithm admits an asynchronous implementation – see, e.g., [16]. In our node-wake-up based asynchronous setting, the same behavior can be achieved if each node i wakes up with equal probability $p_i^u = \frac{1}{n}$, i.e., using uniform clock rates $r_i = r > 0$ for $i \in \mathcal{N}$. The superscript u stands for the uniform choice of clock rates. Then node i picks the edge (i,j) with conditional probability $p_{j|i}^u = \frac{1}{d_i}$ for all $j \in \mathcal{N}_i$. In this case, the unconditional probabilities are given by

$$P_{ij}^u = p_i^u p_{j|i}^u = \frac{1}{n}\frac{1}{d_i}, \quad P_{ji}^u = p_j^u p_{i|j}^u = \frac{1}{n}\frac{1}{d_j} \quad \forall\ (i,j) \in \mathcal{E}.$$

2) Randomized Gossiping with Effective Resistances: This algorithm is similar to the classic randomized gossiping described above, with the only difference that the edges are sampled with non-uniform probabilities proportional to effective resistances $\{R_{ij}\}_{(i,j)\in\mathcal{E}}$. Indeed, in our node-wake-up based asynchronous setting, each node i wakes up with probability $p_i^r = \frac{\sum_{j\in\mathcal{N}_i} R_{ij}}{2\sum_{(i,j)\in\mathcal{E}} R_{ij}}$, i.e., setting clock rate $r_i = \sum_{j\in\mathcal{N}_i} R_{ij}$ for $i \in \mathcal{N}$, and node i picks (i,j) with conditional probability $p_{j|i}^r = \frac{R_{ij}}{\sum_{j\in\mathcal{N}_i} R_{ij}}$ for all $j \in \mathcal{N}_i$. Effective-resistance based gossiping corresponds to the unconditional probabilities

$$P_{ij}^r = p_i^r p_{j|i}^r = \frac{R_{ij}}{2\sum_{(i,j)\in\mathcal{E}} R_{ij}} = \frac{R_{ij}}{2(n-1)} = P_{ji}^r$$

for all $(i,j)\in\mathcal{E}$ where the third equality follows from Fosters Theorem which says that $\sum_{(i,j)\in\mathcal{E}} R_{ij} = (n-1)$ (see e.g. [41]).

In order for the two different gossiping methods to have the same expected number of node-wakeups in a given time period, one should set $r_i = r = 2(n-1)/n$ for $i \in \mathcal{N}$ within the classic randomized gossiping model; hence, the rate of both Poisson processes will be the same, i.e., $\sum_{i\in\mathcal{N}} r_i = 2(n-1)$.

Suppose the edge (i,j) is picked by node i. We can write the update in Step 5 of the Algorithm 3 as

$$y^{k+1} = W_{ij} y^k \quad \text{where} \quad W_{ij} := I - \frac{(e_i - e_j)(e_i - e_j)^\top}{2}.$$

We also define

$$\overline{W}_P \triangleq \mathbb{E}_P[W_{ij}] = \sum_{i,j\in\mathcal{N}} P_{ij} W_{ij}, \tag{8}$$

which is the expected value of the random iteration matrix W_{ij} with respect to the distribution defined over $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$; hence, $\overline{W}_P \mathbf{1} = \overline{W}_P^\top \mathbf{1} = \mathbf{1}$. It can also be seen that W_{ij} is a doubly stochastic, non-negative and a weakly diagonally dominant matrix for every $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$; therefore, $\overline{W}_P$, which is a convex combination of W_{ij} matrices, is also a doubly stochastic, non-negative and weakly diagonally dominant matrix. It follows then from the Gershgorin's Disc Theorem (see e.g. [42]) that all the eigenvalues of $\overline{W}_P$ are non-negative. Moreover, since $\overline{W}_P$ is a non-negative doubly stochastic matrix, its largest eigenvalue $\lambda_n(\overline{W}_P) = 1$. It can be easily checked that $\overline{W}_P$ also satisfies

$$\overline{W}_P = I - \tfrac{1}{2} D + \tfrac{1}{2}(P + P^\top), \tag{9}$$

where D is a diagonal matrix with i-th entry $D_i \triangleq \sum_{j\in\mathcal{N}_i}(P_{ij} + P_{ji})$. Plugging in P^u and P^r for P in this identity respectively leads immediately to the following Lemma.

Lemma 2. *The matrices $\overline{W}_{P^r} = \mathbb{E}_{P^r}[W_{ij}]$ and $\overline{W}_{P^u} = \mathbb{E}_{P^u}[W_{ij}]$ satisfy the identities*

$$\begin{aligned} \overline{W}_{P^u} &= I - \frac{1}{2} D^u + \frac{P^u + (P^u)^\top}{2}, \\ \overline{W}_{P^r} &= I - \frac{1}{2} D^r + \frac{P^r + (P^r)^\top}{2}, \end{aligned}$$

where D^u and D^r are diagonal matrices satisfying $[D^u]_{ii} := \sum_{j\in\mathcal{N}_i}(P_{ij}^u + P_{ji}^u)$, $[D^r]_{ii} = \frac{1}{(n-1)} R_i$ where $R_i \triangleq \sum_{j\in\mathcal{N}_i} R_{ij}$.

Let $\mathcal{A}(P)$ denote an asynchronous gossiping algorithm with probability matrix P characterized by a set of probabilities $\{p_i\}_{i\in\mathcal{N}}$ and $\{p_{j|i}\}_{j\in\mathcal{N}_i}$ for $i \in \mathcal{N}$ as in (7). The performance of such an algorithm is typically measured by the ε-averaging time defined as:

$$T_{ave}(\varepsilon, P) \triangleq \sup_{y^0\in\mathbb{R}^n} \inf \left\{ k : \mathbb{P}\left(\frac{\|y^k - \bar{y}\mathbf{1}\|}{\|y^0\|} \geq \varepsilon \right) \leq \varepsilon \right\}.$$

We will prove later in Section IV that averaging time of the effective resistance-based consensus $T_{ave}(\varepsilon, P^r)$ improves upon that of classical consensus $T_{ave}(\varepsilon, P^u)$ for some graphs. We note that the number of clock ticks can be converted to absolute time easily with standard arguments (simply dividing k by $\sum_{i\in\mathcal{N}} r_i$ to get the expected time of the k-th tick), e.g., see [16, Lemma 1]. This allows us to use number of (clock ticks) iterations to compare asynchronous algorithms. The following theorem from [16] shows that the second largest eigenvalue of $\overline{W}_P$ determines the averaging time.

Theorem 3 ([16, Theorem 3])**.** *For the symmetric matrix $\overline{W}_P$ defined in* (8)*, the following holds:*

$$0.5 \frac{\log(\varepsilon^{-1})}{\log([\lambda_{n-1}(\overline{W}_P)]^{-1})} \leq T_{ave}(\varepsilon, P) \leq 3 \frac{\log(\varepsilon^{-1})}{\log([\lambda_{n-1}(\overline{W}_P)]^{-1})},$$

where $\lambda_{n-1}(\overline{W}_P)$ is the second largest eigenvalue of $\overline{W}_P$.

This result makes the connection between the convergence time of an asynchronous gossiping algorithm and the spectrum of the expected iteration matrix $\overline{W}_P$. Therefore, in order to compare resistance-based gossiping with classical gossiping introduced in Section III-2, it is sufficient to estimate the second largest eigenvalues of $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$ and compare them. In the next section, we discuss estimating the second largest eigenvalues of $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$ based on the notions of graph conductance and hitting times, when the eigenvalues are not readily available in closed form. We will also discuss some examples for which we can explicitly compute the eigenvalues.

Before we move on to the spectral analysis of these expected iteration matrices, it is worth emphasizing that the matrices $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$ are symmetric and doubly stochastic; therefore, they can both be viewed as the probability transition matrix of a reversible Markov Chain on the graph $\mathcal{G}$, both with a uniform stationary distribution. In the next section, we study gossiping algorithms over *barbell graphs*, which are frequently studied in the literature for the consensus problem as they constitute a worst-case example in terms of mixing properties of random walks [5, Section 5] and the performance of distributed averaging algorithms (see e.g. [4], [43]). Barbell graphs are obtained by connecting two copies of a complete graph $K_{\tilde{n}}$ with a single edge (see Figure 1), and are denoted as $K_{\tilde{n}} - K_{\tilde{n}}$.

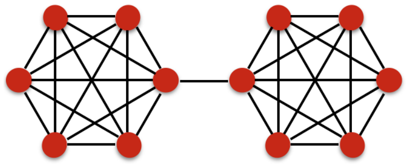

Fig. 1: Barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with $n = 2\tilde{n} = 12$ nodes

For $K_{\tilde{n}} - K_{\tilde{n}}$, let (i^*, j^*) be the edge that connects the two complete subgraphs. This is the only edge that allows node values to be propagated between the complete subgraphs; therefore, how frequently it is sampled is a key factor that determines the averaging time. For example, the unconditional probability of sampling this edge, which we refer to as the *bottleneck edge*, with uniform weights can be computed explicitly as

$$P^u_{i^*j^*} = \frac{1}{n}\frac{1}{d_{i^*}} = \frac{2}{n^2} = P^u_{j^*i^*}. \tag{10}$$

However, we have

$$P^r_{i^*j^*} = P^r_{j^*i^*} = \frac{R_{i^*j^*}}{2(n-1)} = \frac{1}{2(n-1)}, \tag{11}$$

where we used the fact that $R_{i^*j^*} = 1$ (see also the proof of Lemma 16). Comparing the probabilities (10) and (11), we observe that effective resistance weights allow sampling of the bottleneck edge more frequently (by a factor of $\Theta(n)$). A similar effect can also be observed for the Markov chains defined by the transition probability matrices $\overline{W}_{P^u}$ and $\overline{W}_{P^r}$. In fact, by an explicit computation based on Lemma 2, we get

$$[\overline{W}_{P^u}]_{i^*j^*} = [\overline{W}_{P^u}]_{j^*i^*} = \frac{2}{n^2}, \tag{12a}$$

$$[\overline{W}_{P^r}]_{i^*j^*} = [\overline{W}_{P^r}]_{j^*i^*} = \frac{1}{2(n-1)}. \tag{12b}$$

Notice that the probability of moving from one complete subgraph to another is significantly larger (by a factor of $\Theta(n)$) for the effective resistance-based Markov chain. Intuitively speaking, this fact allows the effective resistance-based chain to traverse between two complete subgraphs faster when n is large which in turn leads to faster averaging over the nodes. This will be formalized and proven in the next section.

IV. Theoretical Guarantees for Effective Resistance-Based Consensus

A. Conductance-based analysis

Probability transition matrices on graphs have been studied well in the literature; in particular, there are some combinatorial techniques to bound their eigenvalues based on *graph conductance* [5] as well as some algebraic techniques that allow one to compute all the eigenvalues explicitly exploiting symmetry groups of a graph [44] as we shall discuss in Section IV-B.

The notion of graph conductance is tied to a transition matrix W over a graph which corresponds to a reversible Markov chain admitting an arbitrary stationary distribution π. It can be viewed as a measure of how hard it is for the Markov chain to go from a subgraph to its complement in the worst case. Roughly speaking, low conductance means that there exists a subset of nodes that is not well-connected with the rest of the graph. The notion of graph conductance allows us to provide bounds on the mixing time of the corresponding Markov chain as we discuss below.

Definition 4 (Conductance)**.** *Let W be the transition matrix of a reversible Markov chain*[2] *on the graph $\mathcal{G}$ with a stationary distribution $\pi = \{\pi_i\}_{i=1}^n$. The conductance Φ is defined as*

$$\Phi(W) = \min_{S \subset V} \frac{\sum_{i\in S,\ j\in S^c} \pi_i W_{ij}}{\min\{\pi(S), \pi(S^c)\}}, \tag{13}$$

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

Given the transition matrix W, the relation between conductance $\Phi(W)$ and the second largest eigenvalue $\lambda_{n-1}(W)$ is well-known and given by the *Cheeger inequalities* (see e.g. [45, Proposition 6])

$$1 - 2\Phi(W) \le \lambda_{n-1}(W) \le 1 - \Phi^2(W). \tag{14}$$

Therefore, larger conductance leads to faster averaging time in light of Theorem 3. In particular, we can get lower and upper bounds on the averaging time for both classic and effective resistance based gossiping using the Cheeger's inequality.

Next, we study gossiping algorithms over *barbell graphs* through conductance analysis. In particular, our next result for a barbell graph with $n = 2\tilde{n}$ nodes shows $\Theta(n)$ improvement on the conductance of effective resistance-based transition probabilities $\overline{W}_{P^r}$ compared to uniform probabilities $\overline{W}_{P^u}$.

Proposition 5. *Consider the two Markov chains on the barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ defined by the transition matrices $\overline{W}_{P^u}$ and $\overline{W}_{P^r}$. Their respective graph conductance values are given by*

$$\Phi(\overline{W}_{P^u}) = \frac{4}{n^3}, \quad \Phi(\overline{W}_{P^r}) = \frac{1}{n(n-1)}.$$

By taking the logarithm of the Cheeger inequalities (14), for $\Phi(W) \le 1/2$, we obtain

$$-\log(1 - \Phi^2(W)) \le \log(\lambda_{n-1}^{-1}(W)) \le -\log(1 - 2\Phi(W)). \tag{15}$$

Then, choosing $W = \overline{W}_{P^u}$ and $W = \overline{W}_{P^r}$ above, applying Theorem 3 and Proposition 5 and noting $-\log(1-x) \approx x$ for x close to 0, leads to the following lower and upper bounds on the averaging time of effective resistance-based gossiping and classical gossiping algorithms.

Corollary 6. *The ε-averaging times of gossiping algorithms $\mathcal{A}(P^u)$ and $\mathcal{A}(P^r)$ on the barbell graph with n nodes admit the following bounds:*

$$\Theta(n^3 \log(1/\varepsilon)) \le T_{ave}(\varepsilon, P^u) \le \Theta(n^6 \log(1/\varepsilon)), \tag{16}$$

$$\Theta(n^2 \log(1/\varepsilon)) \le T_{ave}(\varepsilon, P^r) \le \Theta(n^4 \log(1/\varepsilon)). \tag{17}$$

These bounds from Corollary 6 for the barbell graph show that using effective resistances, one can improve upper and lower bounds on the averaging times for consensus by a factor of $\Theta(n)$ and $\Theta(n^2)$, respectively, at the same precision $\epsilon > 0$. One can also consider other similar graphs with low conductance, e.g., c-barbell graph (denoted by $c - K_{\tilde{n}}$) for $c \ge 2$ is a path of c equal-sized complete graphs ($K_{\tilde{n}}$) [46] as illustrated in Figure 2.

For c-barbell graphs, we can show similar improvement for the upper and lower bounds in terms of scaling with respect to the total number of nodes $n = c\tilde{n}$.

[2] That is $\pi(i)W_{ij} = \pi(j)W_{ji}$ for all $i, j \in \mathcal{N}$.

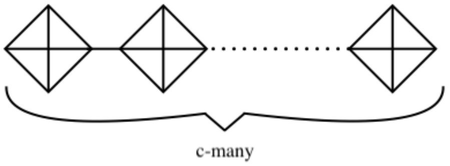

Fig. 2: A c-barbell graph with $\tilde{n} = 4$ (total number of nodes $n = \tilde{n}c = 4c$).

Proposition 7. *Consider the two Markov chains on the c-barbell graph with $n = \tilde{n}c$ nodes defined by the transition matrices $\overline{W}_{P^u}$ and $\overline{W}_{P^r}$. Their respective graph conductance values satisfy*

$$\Phi(\bar{W}_{P^u}) = \frac{c_*}{c\tilde{n}^3}, \quad \Phi(\bar{W}_{P^r}) = \frac{c_*}{2\tilde{n}(c\tilde{n}-1)}, \tag{18}$$

where $c_* = \left(\lfloor \frac{c}{2} \rfloor\right)^{-1}$. Then, it follows that

$$\Theta(c^2\tilde{n}^3) \le T_{ave}(\varepsilon, P^u) \le \Theta(c^4\tilde{n}^6), \tag{19}$$

$$\Theta(c^2\tilde{n}^2) \le T_{ave}(\varepsilon, P^r) \le \Theta(c^4\tilde{n}^4). \tag{20}$$

Although this analysis is also applicable to other graphs with low conductance, it does not typically lead to tight estimates, i.e., the lower and upper bounds do not match in terms of their dependency on n. In the next section, we show that for the case of barbell graphs we get tight estimates on the averaging time by computing the eigenvalues of the averaging matrices $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$ explicitly. More precisely, we will show in Proposition 9 that the lower bounds in (16)–(17) are tight in the sense that $T_{ave}(\varepsilon, P^u) = \Theta(n^3)$ and $T_{ave}(\varepsilon, P^r) = \Theta(n^2)$ and the effective resistance-based averaging is faster by a factor of $\Theta(n)$.

B. Spectral analysis

Eigenvalues of probability transition matrices defined on barbell graphs are studied in the literature. Consider the edge-weighted barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with $n = 2\tilde{n}$ nodes. Let (i^*, j^*) be the edge that connects the two complete subgraphs. The result [44, Prop. 5.1] gives an explicit formula for the eigenvalues of a probability transition matrix W with transition probabilities proportional to edge weights, i.e., $W_{ij} = w_{ij}/\sum_{j\in\mathcal{N}_i} w_{ij}$ where w_{ij} satisfy the following assumptions: $w_{i^*i^*} = w_{j^*j^*} = 0$, $w_{i^*j^*} = A$, $w_{i^*j} = w_{j^*i} = B$ for all $j \in \mathcal{N}_{i^*} \setminus \{j^*\}$ and $i \in \mathcal{N}_{j^*} \setminus \{i^*\}$, $w_{ij} = C$ for all (i,j) in each $K_{\tilde{n}}$ such that $i \neq j$ and $i,j \notin \{i^*, j^*\}$, and $w_{ii} = D$ for $i \in \mathcal{N} \setminus \{i^*, j^*\}$ for some A, B, C, D. However, in our case, all of the entries on the diagonal of $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$ are strictly positive (breaking the $w_{i*i*} = w_{j*j*} = 0$ assumption); therefore, [44, Prop. 5.1] is not directly applicable. In Proposition 8, we adapt this result to our setting with minor modifications to allow $w_{i^*i^*} = w_{j^*j^*} = G$ for any $G > 0$ so that it becomes applicable to $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$. The proof steps are similar to the proof of [44, Prop. 5.1] and are based on exploiting the symmetry properties of the barbell graph illustrated in Figure 3 for $\tilde{n} = 4$. That said, we still provide the modified proof in the supplementary file for the sake of completeness.

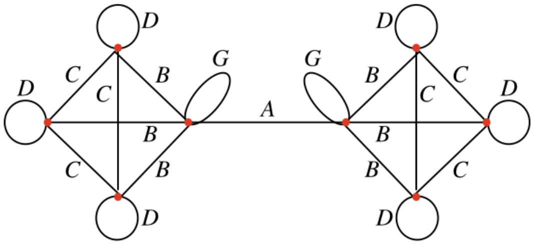

Fig. 3: An edge-weighted barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with edge weights $A, B, C, D, G > 0$ for $\tilde{n} = 4$.

Proposition 8 (Generalization of Proposition 5.1 in [44])**.** *Consider the edge-weighted barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with $n = 2\tilde{n}$ nodes. Let (i^*, j^*) be the edge that connects the two complete subgraphs. Assume that weights are of the form $w_{i^*i^*} = w_{j^*j^*} = G$, $w_{i^*j^*} = A$, $w_{i^*j} = w_{j^*i} = B$ for all $j \in \mathcal{N}_{i^*} \setminus \{j^*\}$ and $i \in \mathcal{N}_{j^*} \setminus \{i^*\}$, $w_{ij} = C$ for all (i,j) in each $K_{\tilde{n}}$ such that $i \neq j$ and $i,j \notin \{i^*, j^*\}$, and $w_{ii} = D$ for $i \in \mathcal{N} \setminus \{i^*, j^*\}$ for some $A, B, C, D, G > 0$. Consider the transition matrix W associated to this graph with entries $W_{ij} = w_{ij}/\sum_{j\in\mathcal{N}_i} w_{ij}$, then the eigenvalues of W are*

- *$\lambda_a := 1$ with multiplicity one,*
- *$\lambda_b := -1 + \frac{A+G}{A+G+E} + \frac{F}{F+B}$ with multiplicity one,*
- *$\lambda_c := \frac{D-C}{B+F}$ with multiplicity $n-4$,*
- *$\lambda_\pm := \frac{1}{2}\Big(\frac{F}{B+F} + \frac{G-A}{A+E+G} \pm \sqrt{S}\Big)$,*

where $E := (\tilde{n}-1)B$, $F := D + (\tilde{n}-2)C$ and $S := \big(\frac{F}{B+F} + \frac{G-A}{A+E+G}\big)^2 - \frac{4(FG-BE-AF)}{(B+F)(A+E+G)}$.

Proof. Proof of Proposition 8 is given in the supplementary material with some background material on the symmetry groups of the barbell graph. □

Based on this result, we characterize the second largest eigenvalue of the consensus matrices $\overline{W}_{P^u}$ and $\overline{W}_{P^u}$ with an explicit computation; next, using Theorem 3 we show that averaging time with resistance weights is $\Theta(n)$ faster on a barbell graph. The proof can be found in the appendix.

Proposition 9. *Consider Markov chains on the barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with transition matrices $\overline{W}_{P^r}$ and $\overline{W}_{P^u}$. The second largest eigenvalues of these matrices are given by*

$$\lambda_{n-1}(\overline{W}_{P^r}) = 1 - \Theta(\frac{1}{n^2}), \quad \lambda_{n-1}(\overline{W}_{P^u}) = 1 - \Theta(\frac{1}{n^3}).$$

Then, it follows from Theorem 3 that

$$T_{ave}(\varepsilon, P^r) = \Theta(\frac{1}{n}) T_{ave}(\varepsilon, P^u).$$

C. Hitting Times and Mixing Times

Before giving a formal definition of the ε-mixing time, we introduce the total variation (TV) distance between two probability measures p and q defined on the vertex set $\mathcal{N} = \{1, 2, \ldots, n\}$. Let p and q be probability measures on $\mathcal{N}$. TV distance between p and q is defined as

$$\|p - q\|_{TV} := \frac{1}{2}\|p - q\|_1.$$

Given a Markov chain with probability transition matrix W and stationary distribution π, ε-mixing time is a measure of how many iterations are needed for the probability distribution of the chain to be ε-close to the stationary distribution with respect to the TV distance.

Definition 10. *Given a Markov chain with probability transition matrix W and stationary distribution π, the ε-mixing time is defined as*

$$T_{mix}(\varepsilon, W) \triangleq \inf_{k\geq 0}\left\{\sup_{p_0\geq 0:\|p_0\|_1=1} \|(W^k)^\top p_0 - \pi\|_{TV} \leq \varepsilon\right\}.$$

Mixing-times and averaging times are closely related. In fact, given probability transition matrix W, it is known that $T_{ave}(\varepsilon, W)$ and $T_{mix}(\varepsilon, \tilde{W})$ admit the same bounds up to $n\log n$ factors [16, Theorem 7] for $\tilde{W} = \frac{I+W}{2}$ – note [16, Theorem 7] uses absolute time whereas we used number of node wakeups to define ϵ-averaging and ϵ-mixing times; therefore, we multiplied $\log(n)$ factor in [16, Theorem 7] by $\sum_{i\in\mathcal{N}} r_i = 2(n-1)$ to convert absolute times to number of node wakeups. Hence, designing algorithms with a smaller mixing time, often leads to better algorithms for distributed averaging (see also [47]). It is also known that mixing time is closely related to hitting times [48, Theorem 1.1] for a Markov Chain where the hitting time $H_W(i \to j)$ is defined as the expectation of the first time that the random walk with transition matrix W reaches j starting from i.

Given a connected $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, suppose there are no self-loops, i.e., $(i,i) \notin \mathcal{E}$ for $i \in \mathcal{N}$. Uniform weights $p^u_{j|i} = \frac{1}{d_i}$ can result in slow mixing on some graphs such as the barbell graph (see Proposition 9) or other graphs like lollipop graphs [5] which have both high degree and low degree nodes together. A popular alternative to uniform weights $\{p^u_{j|i}\}_{j\in\mathcal{N}_i}$ for $i \in \mathcal{N}$ is the Metropolis weights $M = [M_{ij}]_{ij}$ where

$$M_{ij} \triangleq \begin{cases} \frac{1}{\max(d_i,d_j)} & \text{if } (i,j) \in \mathcal{E}, \\ 1 - \sum_{j\in\mathcal{N}_i} \frac{1}{\max(d_i,d_j)} & \text{if } i = j, \\ 0 & \text{else,} \end{cases}$$

and its lazy version uses

$$\tilde{M} \triangleq \frac{I+M}{2},$$

which is popular in the distributed optimization practice [49]. The matrix $\tilde{M}$ is symmetric and positive semi-definite, unlike the matrix M which may have negative eigenvalues that can be close to -1 (therefore, it can be problematic for the convergence of distributed algorithms, see e.g. [24]). Combined with uniform wake-up of nodes, this leads to the following wake-up probabilities for the Metropolis weights based system:

$$P^{\tilde{M}}_{ij} = \frac{1}{n}\tilde{M}_{ij},$$

and the associated matrix

$$\overline{W}_{P^{\tilde{M}}} \triangleq \mathbb{E}_{P^{\tilde{M}}}[W_{ij}] = \sum_{ij} P^{\tilde{M}}_{ij} W_{ij}.$$

In particular, for any connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with n nodes, we have the following guarantees from [49, Lemma 2.1] on the lazy Metropolis weights:

$$\max_{i,j\in\{1,2,\ldots,n\}} H_{\tilde{M}}(i \to j) \leq 12n^2,$$

and

$$\lambda_{n-1}(\tilde{M}) \leq 1 - \frac{1}{71n^2}. \tag{21}$$

By (9), we have also

$$\overline{W}_{P^{\tilde{M}}} = (1 - \frac{1}{n})I + \frac{1}{n}\tilde{M}.$$

Therefore, from (21), we get the bound

$$\lambda_{n-1}(\overline{W}_{P^{\tilde{M}}}) \leq 1 - \frac{1}{71n^3},$$

for any connected graph $\mathcal{G}$. Therefore, we conclude from Theorem 3 that the ε-averaging time of Metropolis weights-based gossiping on any graph is $\mathcal{O}(n^3\log(1/\varepsilon))$ – again using the fact that $-\log(1-x) \approx x$ for x close to 0. That said, in the special case of the barbell graph, Metropolis weights perform similar to uniform weights; both require $\Theta(n^3\log(1/\varepsilon))$ time which is improved by the effective resistance-based weights to $\Theta(n^2\log(1/\varepsilon))$. Furthermore, it can be shown that among all the gossiping algorithms $\mathcal{A}(P)$ with a symmetric P matrix on the barbell graph, this rate is optimal with respect to ε and n and cannot be improved[3]. In this sense, effective-resistance based gossiping with $P = P^r$ leads to optimal performance on the barbell graph with respect to scaling in n and ε among all the symmetric choices of the P matrix.

Next, given any connected graph $\mathcal{G}$, we obtain a bound on the second largest eigenvalue of the $\overline{W}_{P^r}$ and show that the averaging time with effective resistance weights $T_{ave}(\varepsilon, P^r) = \mathcal{O}\left(\mathcal{D}n^3\log(1/\varepsilon)\right)$ where $\mathcal{D}$ is the diameter[4] of the graph. In fact, if the diameter $\mathcal{D} \leq 11$, our bounds for the effective-resistance weights improve upon that of Metropolis chain by a (small) constant factor. This would for instance hold for barbell graphs where $\mathcal{D} = 3$ or for mid-size *small world* graphs which are random graphs that arise frequently in real-world applications where nodes can be reached from every other node by a small number of steps [51][5].

Theorem 11. *Let $\mathcal{G}$ be a graph with diameter $\mathcal{D}$. The second largest eigenvalue of the $\overline{W}_{P^r}$ matrix satisfies*

$$\lambda_{n-1}(\overline{W}_{P^r}) \leq 1 - \frac{1}{6\mathcal{D}n^3}.$$

[3] Note that when P is symmetric, it is doubly stochastic. For large n and doubly stochastic P, by [16, Corollary 1], we have $T_{ave}(\varepsilon, P) = \Theta\left(\frac{n\log(1/\varepsilon)}{1-\lambda_2(P)}\right)$. On the other hand, Roch proved that [50, Section 3.3.1.] any doubly stochastic P matrix on the barbell graph with n nodes satisfies the bound $\frac{1}{1-\lambda_2(P)} = \Omega(n)$. Therefore, we obtain $T_{ave}(\varepsilon, P) = \Omega\left(n^2\log(1/\varepsilon)\right)$.

[4] The diameter of a graph is $\mathcal{D} = \max_{i,j\in\mathcal{N}} d(i,j)$ where $d(i,j)$ is the shortest path on the graph between nodes i and j.

[5] For example, we observe that the small world graphs that we randomly created with parameters $n = \{5, 10, 15, 20, 25\}$ and $m = \lfloor 0.2(n^2 - n)\rfloor$ using the methodology described in the numerical experiments (see Section V-A) satisfy $\mathcal{D} \leq 5$ on average over 10^4 independent and identically distributed (i.i.d.) samples. Alternative construction techniques for these graphs are also studied by Cont *et al.* [51] where it is shown that randomized community-based small world graphs admit $2\log(n)$ upper bound almost surely on the diameter diameter $\mathcal{D}$. For instance, these graphs will satisfy $\mathcal{D} \leq 11$ almost surely for $n \leq 240$.

Proof: It follows from our discussion in Section III-2 that $\overline{W}_{Pr}$ is non-negative and doubly stochastic (see the paragraph before Lemma 2). Therefore, for analysis purposes, we can interpret $\overline{W}_{Pr}$ as the transition matrix of a Markov chain $\mathcal{M}$ whose stationary distribution π is the uniform distribution. Our analysis is based on relating the eigenvalues of $\overline{W}_{Pr}$ matrix to the hitting times of the Markov chain $\mathcal{M}$ where we follow the proof technique of [49, Lemma 2.1]. By Lemma 17 from the appendix, we get

$$H_{\overline{W}_{Pr}}(i \to j) \leq n\frac{2(n-1)}{R_{ij}} \quad \text{if} \quad j \in \mathcal{N}_i.$$

For any graph, it is also known that[6]

$$\min_{i,j} R_{ij} \geq \frac{2}{n}.$$

Therefore, for any neighbors i and j,

$$H_{\overline{W}_{Pr}}(i \to j) \leq n^2(n-1).$$

For any two vertices i and j not necessarily neighbors, $i \neq j$, let $v_0(=i), v_1, \ldots, v_\ell(=j)$ be a shortest path connecting i and j. Then, by the sub-additivity property of hitting times, for any $i, j \in \mathcal{N}$, we obtain

$$H_{\overline{W}_{Pr}}(i \to j) \leq \ell n^2(n-1) \leq \mathcal{D}n^2(n-1).$$

It follows from an analysis similar to [53] that

$$\begin{aligned} T_{mix}(\frac{1}{8}, \overline{W}_{Pr}) &\leq 8 \max_{i,j \in \{1,\ldots,n\}} H_{\overline{W}_{Pr}}(i \to j) + 1 \\ &\leq 8\mathcal{D}n^3. \end{aligned} \tag{22}$$

From [53, eqn. (12.12)], we also have

$$T_{mix}(\frac{1}{8}, \overline{W}_{Pr}) \geq \left(\frac{1}{1-\lambda_{n-1}(\overline{W}_{Pr})} - 1\right)\ln(4).$$

Combining this with the estimate (22) implies directly

$$\lambda_{n-1}(\overline{W}_{Pr}) \leq 1 - \frac{1}{6\mathcal{D}n^3},$$

which proves the claim. ■

V. Numerical Experiments

In this section, first we provide numerical experiments to show that $\{R_{ij}\}_{(i,j)\in\mathcal{E}}$ can be computed very efficiently in a decentralized fashion for a given connected $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; second, we demonstrate the benefits of using effective resistances for both solving the consensus problem and also within DPGA-W [27] and EXTRA [24] algorithms for consensus optimization.

A. Decentralized computation of $\mathcal{L}^\dagger$

We tested D-RK, shown in Algorithm 2, and its normalized version, i.e., D-RK on $S^{-1/2}\mathcal{L}X = S^{-1/2}B$, for unweighted small-world-type communication networks, and we compared these randomized methods with deterministic (cyclic) Kaczmarz method. Given positive integers n, m such that $m \geq n$, let $E \in \mathbb{S}^n$ denote the adjacency matrix of the small-wold network parameterized by (n, m) such that $E_{i,i+1} = 1$ for $i = 1, \ldots, n-1$ and $E_{1,n} = 1$, and the other $m - n$ entries are chosen uniformly at random among the remaining upper diagonal elements of E and set to 1. We considered $n \in \{10, 20\}$ and for each n, we chose m such that the edge density, $2m/(n^2 - n)$, is 0.4 or 0.8. For each scenario, we plot the average of $\log\log(1 + \|X^k - \mathcal{L}^\dagger\|_F / \|\mathcal{L}^\dagger\|_F)$ over 100 sample paths versus iteration counter k, i.e, number of node wake-ups.

The results show that the randomized algorithms are slower than their deterministic counterpart; this is the price to pay for asynchronous computations. D-RK applied to the *normalized system* was also faster than the standard D-RK, i.e., numerically we see $\rho_S < \rho$ as suggested by the inequality (6) and Proposition 1. We also observed *finite convergence* on every sample path numerically – the finite number of iterations required for convergence depended on the sample path chosen; hence, averaging iterates over sample paths led to the smooth curves reported in Fig. 4. Given $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with n nodes, m edges, and some $b \in \mathbb{R}^n$, $\tilde{\mathcal{O}}(m)$ convergence behavior has already been observed for randomized Kaczmarz (RK) methods to solve Laplacian systems $\mathcal{L}x = b$ where $\tilde{\mathcal{O}}$ hides some logarithmic factors in m, i.e., to compute $\mathcal{L}^\dagger b$ – this would naively imply $\tilde{O}(mn)$ RK iterations for computing $\mathcal{L}^\dagger$, e.g., see [54]. This might explain why numerically we observe finite convergence in our tests.

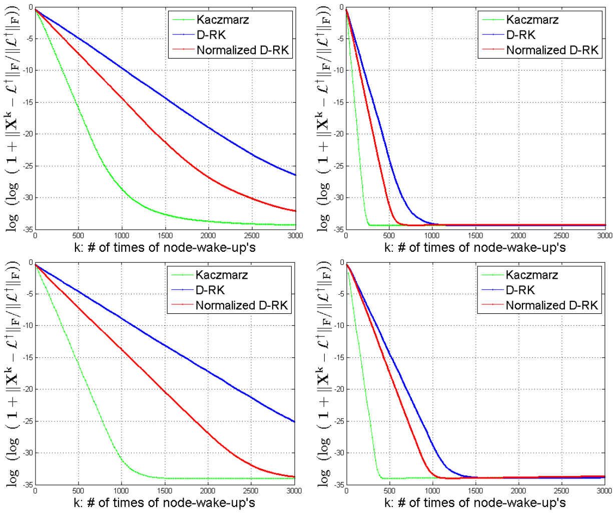

Fig. 4: Performance of D-RK and normalized D-RK on small-world $\mathcal{G}$: **top, left:** $(n, m) = (10, 18)$, **top, right:** $(n, m) = (10, 36)$, **bottom, left:** $(n, m) = (20, 76)$, **top, right:** $(n, m) = (20, 152)$.

[6] This follows directly from the Rayleigh's mononotonicity rule [6] which says that if an edge is removed from a graph, effective resistance on any edge can only increase. Therefore, the complete graph provides a lower bound for R_{ij} where $R_{ij} = 2/n$ (see also [52]).

B. Consensus exploiting effective resistances

1) Classic vs ER based randomized gossiping: We compare the performance of both protocols over an unweighted barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with $n = 2\tilde{n}$ nodes. Such a graph is illustrated in Fig. 1. In our experiment, we set $\tilde{n} = 20$. Let $\mathcal{N}_R = \{1, \dots, 20\}$ and $\mathcal{N}_L = \{21, \dots, 40\}$ represent the node sets in right and left complete subgraphs ($K_{\tilde{n}}$ on the right and left) of the barbell graph.

To initialize y^0, we sample y_i^0 from $\mathbf{N}(100, 1)$ for $i \in \mathcal{N}_L$ and y_i^0 from $\mathbf{N}(0, 1)$ for $i \in \mathcal{N}_R$ where $\mathbf{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2 – this way both lobes have significantly different local means. On the left of Fig. 5, we plot $\log\log(1 + \left\|y^k - \bar{y}\mathbf{1}\right\|_2 / |\bar{y}|)$; and on the right, we plot $\frac{1}{20}\sum_{i\in\mathcal{N}_L} y_i^k$ and $\frac{1}{20}\sum_{i\in\mathcal{N}_R} y_i^k$ vs k for both protocols. The results show that randomized gossiping with effective resistances is much faster.

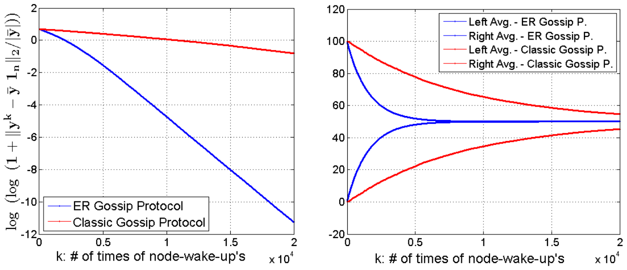

Fig. 5: Performance of classic vs effective resistance based gossiping on barbell $K_{20} - K_{20}$: **left:** Relative error vs k, **right:** Average of left and right lobes vs. k for both protocols.

2) Effective resistance-based gossiping vs state-of-the-art: Gossiping algorithms have been studied vastly in the literature and there have been a number of approaches [47], [55], [56], [57], [58], [59], [60]. In light of Theorem 3, the fastest gossiping algorithm $\mathcal{A}(P)$ is obtained for the choice of P that minimizes the second largest eigenvalue $\lambda_{n-1}(\overline{W}_P)$. Boyd *et al.* show that such an optimal probability matrix P^{opt} can be computed by solving a semi-definite programming (SDP) problem and develop a distributed subgradient method to compute it [16, Section IV]. This method requires a decaying stepsize and computation of the subgradient of the objective $\lambda_{n-1}(\overline{W}_P)$ with respect to the decision variable P at every iteration which itself requires solving a consensus problem at every iteration. This can be expensive in practice in terms of average number of communications required per node. Furthermore, as the authors mention in [16], the subgradient methods are relatively slow with at most sublinear convergence behavior due to the decaying stepsize (compared to a linearly convergent algorithm such as the D-RK method) and has no simple stopping criterion that guarantees a certain level of suboptimality. The resulting gossiping algorithm with optimal choice of the probability matrix P^{opt} is called the Fastest Mixing Markov Chain (FMMC) in the literature [61]. Our algorithm, the effective resistance-based gossiping algorithm (ER), computes a probability matrix P^r faster (with linear convergence guarantees). Even-though the resulting P^r probabilities are not optimal, in the following numerical experiments we illustrate that ER works well in practice in terms of total number of communications required.

In our experiments, we compare average communication times per node of ER and FMMC on barbell graphs and small world graphs. This comparison consists of two stages: (i) pre-computation stage (where the probability matrices P^r and P^{opt} are computed up to a given tolerance) (ii) asynchronous consensus stage (where we run ER and FMMC with probability matrices P^r an P^{opt} obtained from the previous stage to solve a consensus problem). Since FMMC does not have a simple stopping criterion, we first compute the optimal probability matrix P^{opt} with a centralized algorithm based on interior-point methods using the CVX software [62]. Similarly, we compute $\mathcal{L}^\dagger$ for ER. Then, we stop the distributed subgradient method to compute P^{opt} when the iterate P_k at step k of the method satisfies $\frac{||P_k - P^{opt}||_F}{||P^{opt}||_F} \leq \epsilon_1$ and ϵ_1 is the given precision level. We also stop the D-RK algorithm similarly when the iterate X^k at step k satisfies $\frac{\|X^k - \mathcal{L}^\dagger\|_F}{\|\mathcal{L}^\dagger\|_F} \leq \epsilon_1$. In the second (asynchronous consensus) stage, at every iteration, a node contacts a neighbor to compute an average of their state vectors, this is considered as *one communication* for the node. For the barbell graph, the initial state vector $y_i^0 \in \mathbb{R}^4$ for consensus and is sampled from the normal distribution $\mathbf{N}(500, 10)$ if $i \in \mathcal{N}_L$ and from $\mathbf{N}(-500, 10)$ if $i \in \mathcal{N}_R$. We stop the consensus stage for both algorithms when the iterates y^k satisfy the error $\frac{||y^k - \bar{y}||}{\sqrt{p}||\bar{y}||} \leq \epsilon_2$ where ϵ_2 is the tolerance level.

Graph	Method	Ave. comm. per node (pre-comp.) (x10^3)	Ave. comm. per node (asynch. consensus)	Tot. ave. comm. per node (x10^3)
$K_5 - K_5$	ER FMMC	1.5 8.1	81 65	1.5 8.1
$K_{10} - K_{10}$	ER FMMC	42.5 90.6	197.67 129.60	42.7 90.7
$K_{15} - K_{15}$	ER FMMC	313.4 511.4	313.31 189.85	313.8 511.6
$K_{20} - K_{20}$	ER FMMC	1299.9 10170.3	432.98 251.15	1300.3 10170.5
$K_{25} - K_{25}$	ER FMMC	3926.1 >33x10^{10}	565.96 287.30	3926.7 >33x10^{10}

TABLE I: The comparison of FMMC and ER on the barbell graph

Table I compares the average number of communications per node required for $\epsilon_1 = \epsilon_2 = 0.01$ accuracy for both FMMC and ER on the barbell graph. We use the recommended decaying stepsize $\alpha_k = R/k$ from [16] for the distributed subgradient method where the stepsize amplitude parameter R is tuned to the graph to achieve the best performance. Results are reported in Table I in which we compare the average communication per node in the pre-computation stage (third column of I) and in the second (asynchronous consensus) stage (fourth column of I) as well as a cumulative sum of both stages (fifth column of I) when the number of nodes is varied. We observe that consistently ER requires significantly less communication per node in the pre-computation stage. This is because at each step of the pre-computation, FMMC computes a subgradient of the objective $\lambda_{n-1}(\overline{W}_P)$ with respect to P which requires solving a consensus problem at every iteration, and this can be expensive in terms of total number of communications required. Since FMMC finds the optimal probabilities, the asynchronous consensus stage with FMMC and target accuracy of $\epsilon_2 = 0.1$ requires less communications by a factor of at most two. However, when we consider the cumulative sum of both stages, we observe that ER requires significantly less communication.

Next, we compare ER and FMMC on small world graphs in Table II. We took small world graphs that have an edge density of $\frac{2m}{n^2-n} \approx 0.4$ (by choosing $m = \lfloor 0.2(n^2 - n) \rfloor$) as the number of nodes n is varied and similar to previous experiments we compared the pre-computation stage, asynchronous consensus stage and the cumulative sum of both stages in terms of total number of communications per node with tolerance level $\epsilon_1 = \epsilon_2 = 0.05$. In the precomputation stage, we terminate both algorithms (D-RK and the distributed subgradient method) if the target accuracy $\epsilon_1 = 0.05$ is achieved or if the average number of communications per node reach out to a maximum limit of 10^6. ER required significantly less communications per node than the maximum communication limits to operate, whereas FMMC algorithm reached the maximum communication limit when the number of nodes was 10 or larger. In the second (asynchronous consensus) stage, we used the calculated probabilities that approximate P^r and P^{opt} from the pre-computation stage within the target accuracy ε_1 and the maximum communication limit. We observe that ER performs consistently better than FMMC in the second stage as well. Finally, the last column of Table II reports the average communication required per node to complete both stages, and we see that ER requires significantly less number of communications. These results show that ER is a practical algorithm in terms of communication efficiency on the small-world graphs as well.

Graph	Method	Ave. comm. per node (pre-comp.)	Ave. comm. per node $\begin{pmatrix}\text{asynch.}\\\text{consensus}\end{pmatrix}$	Tot. ave. comm. per node
$n = 5$	ER FMMC	6.4 11160.8	40.53 84	47 11.245
$n = 10$	ER FMMC	16.8 $> 10^6$	130.30 142.84	147 $> 10^6$
$n = 15$	ER FMMC	14.93 $> 10^6$	225.63 271,00	240.56 $> 10^6$
$n = 20$	ER FMMC	19.20 $> 10^6$	315.22 370.00	334.42 $> 10^6$
$n = 25$	ER FMMC	500.00 $> 10^6$	403.32 512.00	903.32 $> 10^6$

TABLE II: The comparison of FMMC and ER on the small world graph

C. Effective resistance-based DPGA-W and EXTRA

We implemented our resistance-based communication framework into the state of the art distributed algorithms: DPGA-W [27] and EXTRA [24] on the regularized logistic regression problem over a barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with $n = 2\tilde{n}$ nodes:

$$\min_{x\in\mathbb{R}^p} \sum_{i=1}^{n} f_i(x), \quad f_i(x) \triangleq \tfrac{1}{2n}\|x\|^2 + \tfrac{1}{N_s}\sum_{\ell=1}^{N_s} \log(1 + e^{-b_{i\ell} a_{i\ell}^\top x}), \tag{23}$$

where N_s is the number of samples at each node, $\{(a_{i\ell}, b_{i\ell})\}_{\ell=1}^{N_s} \subset \mathbb{R}^p \times \{-1, 1\}$ for $i \in \mathcal{N}$ denote the set of feature vectors and corresponding labels. We let $p = 20$ and $N_s = 5$. For each $n \in \{20, 40\}$ and $\sigma \in \{1, 2\}$, we randomly generated 20 i.i.d. instances of the problem in (23) by sampling $a_{i\ell} \sim \mathbf{N}(\mathbf{1}, \sigma^2\mathbf{I})$ independently from the normal distribution and setting $b_{i\ell} = -1$ if $1/(1 + e^{-a_{i\ell}^\top \mathbf{1}}) \leq 0.55$ and to $+1$ otherwise. Both algorithms are terminated after 10^4 iterations. For benchmark, we also solved each instance of (23) using MOSEK [63] within CVX [62]. We initialized the iterates uniformly sampling each p components from the $[500, 510]$ interval for nodes in one $K_{\tilde{n}}$, and from $[-500, -490]$ for nodes in the the other $K_{\tilde{n}}$. The results for $n = 20$ and $n = 40$ are displayed in Fig. 6 and Fig. 7, respectively. We plotted relative suboptimality $\|\mathbf{x}^k - \mathbf{x}^*\| / \|\mathbf{x}^*\|$, function value sequence $\sum_{i\in\mathcal{N}} f_i(x_i^k)$ for the range $[0,\ 10^5]$, and consensus violation $\|\mathbf{x}^k - \bar{\mathbf{x}}^k\| / \sqrt{n}$, where k denotes the (synchronous) communication round counter – in each communication round neighboring nodes communicate among each other synchronously once – and $\mathbf{x}^k = [x_i^k]_{i\in\mathcal{N}}$ denotes the k-th iterate; moreover, $\bar{\mathbf{x}}^k = \mathbf{1} \otimes \bar{x}^k$, $\bar{x}^k = \sum_{i\in\mathcal{N}} x_i^k / n$, $\mathbf{x}^* := \mathbf{1} \otimes x^*$ and x^* is the minimizer to (23).

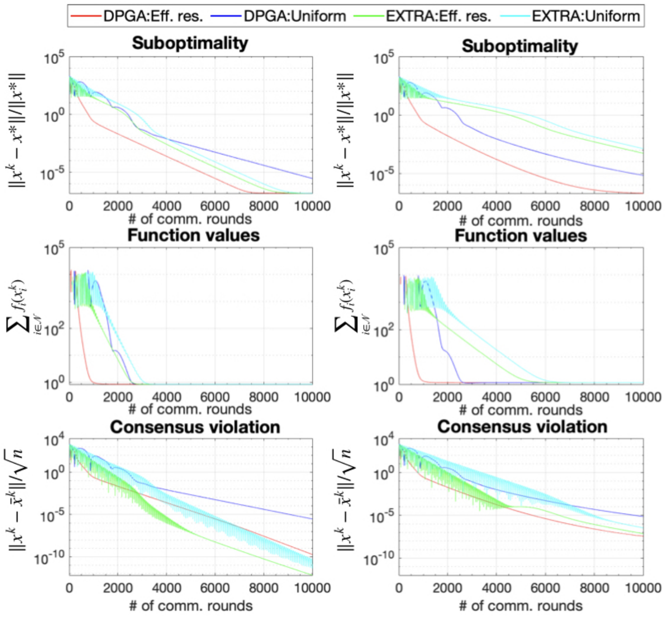

Fig. 6: The suboptimality, function value and difference from average comparison of logistic regression using DPGA-W and EXTRA algorithms with resistance weights and uniform probability weights on barbell graph $K_{10} - K_{10}$. **Left:** Data of the logistic regression model is sampled using $\sigma = 1$, **Right:** Data is sampled using $\sigma = 2$.

Both DPGA-W[7] and EXTRA uses a communication matrix W that encodes the network topology. DPGA-W uses node-specific step-sizes initialized at $\approx 1/L_i$ for $i \in \mathcal{N}$, where L_i denotes the Lipschitz constant of ∇f_i, we adopted the adaptive step-size strategy described in [27, Sec. III.D]; and for EXTRA, we choose the constant step-size, common for all nodes, as suggested in [24], i.e., we choose the stepsize as $2\lambda_{\min}(\tilde{W}) / \max_{i\in\mathcal{N}} L_i$, where $\tilde{W} = (\mathbf{I} + W)/2$.

[7] In DPGA-W stepsize parameter γ_i is set to $1/\|\omega_i\|$ for $i \in \mathcal{N}$ – see [27].

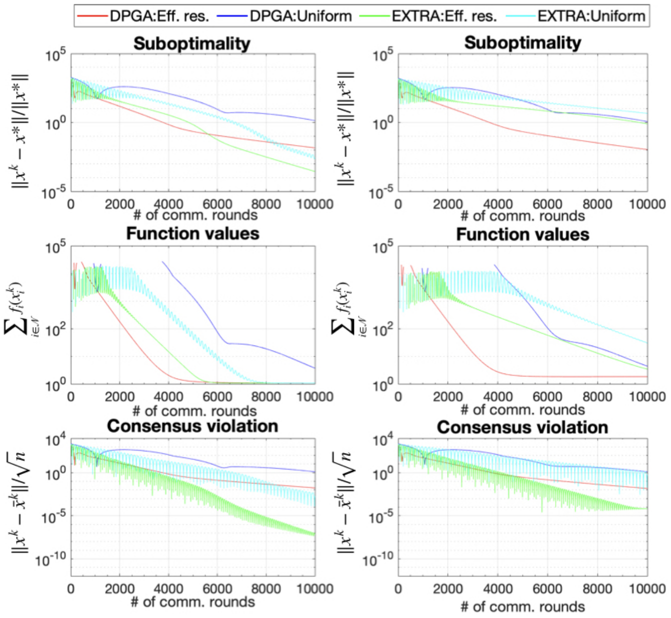

Fig. 7: The suboptimality, function value and difference from average comparison of logistic regression using DPGA-W and EXTRA algorithms with resistance weights and uniform probability weights on the barbell graph $K_{20} - K_{20}$. **Left:** Data of the logistic regression model is sampled using $\sigma = 1$, **Right:** Data is sampled using $\sigma = 2$.

For both algorithms, we compared two choices of W: W^u based on uniform edge weights, and W^r based on effective resistances. In DPGA-W, the graph Laplacian is adopted for uniform weights, i.e., $W^u = W^{u,\text{DPGA-W}} := \mathcal{L}$, while for the effective resistance-based weights, we set $W^r = W^{r,\text{DPGA-W}}$ where $W_{ii}^{r,\text{DPGA-W}} := \sum_{j\in\mathcal{N}_i} R_{ij}$ for $i \in \mathcal{N}$ and $W_{ij}^{r,\text{DPGA-W}} = -R_{ij}$ for $(i,j) \in \mathcal{E}$ and 0 otherwise. For EXTRA, $W^{u,\text{EXTRA}} = \mathbf{I} - \mathcal{L}/\tau$ where $\tau = \lambda_{\max}(\mathcal{L})/2 + \varepsilon$; on the other hand, $W^{r,\text{EXTRA}} = \mathbf{I} - W^{r,\text{DPGA-W}}/\tau$ where $\tau = \lambda_{\max}(W^{r,\text{DPGA-W}})/2 + \varepsilon$ for $\varepsilon = 0.01$.

Figures 6 and 7 give the performance comparison of both DPGA-W and EXTRA algorithms with effective resistance and uniform weights in terms of suboptimality, convergence in function values and consensus violation for the barbell graph $K_{10} - K_{10}$ and $K_{20} - K_{20}$ respectively – the reported results are averages over the 20 problem instances. The left panels of Figures 6 and 7 report noise level $\sigma = 1$ whereas right panels report $\sigma = 2$ (which corresponds to noisier data). In Figures 6 and 7, we observe that effective resistance weights improves upon the uniform weights for both EXTRA and DPGA-W methods consistently to solve the logistic regression problem in terms of suboptimality, function values and consensus violation signficantly. We also observe that with noisier data (when $\sigma = 2$ as opposed to $\sigma = 1$), DPGA-W works typically faster than EXTRA in terms of function values and suboptimality. This is because when noise level σ gets larger, the local Lipschitz constant L_i of the nodes demonstrate higher variability, and DPGA-W adapts to this variability as it uses a stepsize that is different at each node in a way to adapt to L_i, whereas EXTRA uses a constant stepsize that is the same for each node. On the other hand, in terms of consensus violation, we see that EXTRA with effective resistance weights typically outperforms DPGA-W with effective resistance weights.

VI. Conclusions and Future Work

In this work, we developed a linearly convergent distributed algorithm for computing effective resistances over an undirected connected graph $\mathcal{G}$. Our method builds on an efficient, distributed and asynchronous implementation of the Kaczmarz method for solving linear Laplacian systems $\mathcal{L}x = b$. We also presented an application of our algorithm to accelerate distributed optimization algorithms including the EXTRA, DPGA-W and randomized gossiping methods, and both numerically and theoretically demonstrated that our algorithms can accelerate existing state-of-the-art methods.

As part of our future work, we will investigate the finite convergence properties of our algorithm to compute the effective resistances, suggested by the experiments. Finally, we will investigate the applications of effective resistances to a wider class of distributed optimization algorithms beyond EXTRA and DPGA-W such as ADMM or other operator splitting-based distributed methods.

Appendix A
Proof of Proposition 1

We first prove the following lemma which shows that the diagonal entries $\{s_i\}_{i=1}^n$ of S are related to the Frobenius norm of $\mathcal{L}$.

Lemma 12. *The Laplacian $\mathcal{L}$ has the following property:*

$$\frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{s_i} \geq \frac{1}{||\mathcal{L}||_F^2}, \tag{24}$$

where s_i is defined by (3)*.*

Proof: Note that

$$||\mathcal{L}||_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{ij}^2 = \sum_{i=1}^{n}\sum_{j\in N_i\cup\{i\}}\mathcal{L}_{ij}^2 = \sum_{i=1}^{n} s_i,$$

where we used the fact that $\mathcal{L}_{ij} = 0$ for all $(i,j) \notin \mathcal{E}$. Applying arithmetic-harmonic mean inequality to the sequence $\{s_i\}_{i\in\{1,..,n\}}$, we obtain

$$\frac{1}{n}||\mathcal{L}||_F^2 = \frac{1}{n}\sum_{i=1}^{n} s_i \geq n\Big[\sum_{i=1}^{n}\frac{1}{s_i}\Big]^{-1}.$$

Multiplying both sides with $1/n$ leads to (24). ■

Since $\mathcal{L}$ and S are symmetric matrices so are $\mathcal{L}^2$ and S^{-1}. Let $\{\lambda_i(\mathcal{L})\}_{i=1}^n$ and $\{\lambda_i(S)\}_{i=1}^n$ denote the eigenvalues of these matrices sorted in increasing order, i.e. λ_n is the largest eigenvalue, λ_1 is the smallest one. By the eigenvalue interlacing result in [64, Chapter 2, Eq. (2.0.7)], we obtain[8]

$$n\lambda_2(\mathcal{L}^2\mathcal{S}^{-1}) \geq \lambda_2(\mathcal{L}^2)\sum_{i=1}^{n}\lambda_i(\mathcal{S}^{-1}), \tag{25}$$

where all the matrices have non-negative real eigenvalues as both $\mathcal{L}$ and S are symmetric with non-negative eigenvalues. Clearly, $\lambda_2(\mathcal{L}^2) = \lambda_2(\mathcal{L})^2 > \lambda_1(\mathcal{L}^2) = 0$. Furthermore, the eigenvalues of $\mathcal{L}^2\mathcal{S}^{-1}$ and $\mathcal{L}\mathcal{S}^{-1}\mathcal{L}$ are the same because if u

[8] We set $l = n$ and $i_t = 2$ for $t = 1, \ldots, l$ in Eq. (2.0.7) in [64].

is an eigenvector of the latter matrix corresponding to a nonzero eigenvalue λ, then $\mathcal{L}u$ would be the right eigenvector of the former matrix with the same eigenvalue; similarly, if u is a right-eigenvector of $\mathcal{L}^2\mathcal{S}^{-1}$ corresponding to a nonzero eigenvalue λ, then $\mathcal{L}\mathcal{S}^{-1}u$ is an eigenvector of $\mathcal{L}\mathcal{S}^{-1}\mathcal{L}$ with the same eigenvalue. Therefore, since $\mathcal{L}\mathcal{S}^{-1}\mathcal{L}$ is positive semidefinite with $\lambda_1(\mathcal{L}\mathcal{S}^{-1}\mathcal{L}) = 0$, we also have

$$\lambda_1(\mathcal{L}^2\mathcal{S}^{-1}) = 0. \tag{26}$$

Moreover, S is a diagonal matrix with diagonal entries $S_{ii} = s_i$; therefore, eigenvalues of S are given by s_i with $i = 1, 2, \ldots, n$. Hence (25) is equivalent to

$$n\lambda_2(\mathcal{L}^2\mathcal{S}^{-1}) \geq \lambda_2(\mathcal{L})^2 \sum_{i=1}^{n} \frac{1}{s_i} \geq \lambda^+_{\min}(\mathcal{L})^2 \frac{n^2}{\|\mathcal{L}\|_F^2} > 0, \tag{27}$$

where the inequalities follow from Lemma 12 and the fact that $\lambda_2(\mathcal{L}) = \lambda^+_{\min}(\mathcal{L}) > 0$ due to $\mathcal{G}$ being a connected graph, where $\lambda^+_{\min}(\cdot)$ denotes the smallest positive eigenvalue. From (26) and (27), we conclude that $\lambda_2(\mathcal{L}^2\mathcal{S}^{-1})$ is the smallest positive eigenvalue of $\mathcal{L}^2\mathcal{S}^{-1}$, i.e.,

$$\lambda_2(\mathcal{L}^2\mathcal{S}^{-1}) = \lambda^+_{\min}(\mathcal{L}^2\mathcal{S}^{-1}). \tag{28}$$

Finally, using the fact that the eigenvalues of $\mathcal{L}^2\mathcal{S}^{-1}$ and $\mathcal{L}\mathcal{S}^{-1}\mathcal{L}$ are the same once again, we get $\lambda^+_{\min}(\mathcal{L}\mathcal{S}^{-1}\mathcal{L}) = \lambda^+_{\min}(\mathcal{L}^2\mathcal{S}^{-1})$. Combining this with (27) and (28) leads to

$$\frac{1}{n}\lambda^+_{\min}(\mathcal{L}\mathcal{S}^{-1}\mathcal{L}) = \frac{1}{n}\lambda^+_{\min}(\mathcal{L}^2\mathcal{S}^{-1}) \geq \Big(\frac{\lambda^+_{\min}(\mathcal{L})}{\|\mathcal{L}\|_F}\Big)^2,$$

which proves (6). Then, the fact that $\rho_S \leq \rho$ follows directly from the definitions in (3) and (5).

Appendix B
Proofs of Propositions 5 & 7

Proof of Proposition 5: Notice that the stationary distribution π of a Markov chain with a doubly stochastic transition matrix W is uniform, i.e. $\pi(i) = \frac{1}{n}$ for all $i \in \mathcal{N}$; therefore, the sum (13) becomes

$$\Phi(W) = \min_{\substack{S\subset\mathcal{N} \\ |S| \leq \frac{|\mathcal{N}|}{2}}} \frac{1}{|S|} \sum_{i\in S,\, j\in S^c} W_{ij}. \tag{29}$$

For both $W = \overline{W}_{P^u}$ and $W = \overline{W}_{P^r}$, we can compute all the entries of these matrices explicitly (see (12) and Lemma 16). Then, it follows after a straightforward computation that the minimum in (29) is attained when S is the set of the vertices of one of the complete subgraphs of the barbell graph. Since (i^*, j^*) is the only edge connecting S and S^c, it follows from (12) that

$$\begin{aligned} \Phi(\overline{W}_{P^u}) &= \frac{1}{\tilde{n}}[\overline{W}_{P^u}]_{i^*j^*} = \frac{4}{n^3}, \\ \Phi(\overline{W}_{P^r}) &= \frac{1}{\tilde{n}}[\overline{W}_{P^r}]_{i^*j^*} = \frac{1}{n(n-1)}. \end{aligned}$$

■

Proof of Proposition 7: The proof is constructive. Given $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, the conductance of a subset $S \subset \mathcal{N}$ with respect to the probability transition matrix W is defined as

$$\Phi_S(W) := \frac{1}{\pi(S)} \sum_{i\in S, j\in S^C} \pi(i) W_{ij}. \tag{30}$$

With slight abuse of notation, for a subgraph $\mathcal{H}_0$ with a vertex set S_0, we define $\Phi_{\mathcal{H}_0}(W) := \Phi_{S_0}(W)$. Note that by the definition of conductance (13), we have $\Phi(W) = \min_S \Phi_S(W)$.

We say that a vertex set $S \subset \mathcal{N}$ on graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, w)$ is a *one-cut set* if its complement $\mathcal{N} \setminus S$ is a connected subgraph of $\mathcal{G}$. Similarly, we define *two-cut set* $S_2 \subset \mathcal{N}$ to be a set whose complement $\mathcal{N} \setminus S_2$ consists of two disjoint non-empty connected subgraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ of $\mathcal{G}$. We define

$$\mathcal{G}_1 := \text{the left-most clique of the } c\text{-barbell graph.} \tag{31}$$

For $c_0 \in [2, c]$, we also define

$$\begin{aligned} \mathcal{G}_{c_0} := {} & c_0\text{-barbell subgraph that includes the left-most} \\ & c_0 \text{ cliques of the } c\text{-barbell graph,} \end{aligned} \tag{32}$$

For example, $\mathcal{G}_2$ is the barbell graph that contains the left-most two complete subgraphs; whereas $\mathcal{G}_c$ is the c-barbell graph. Notice that matrices W_{P^u} and W_{P^r} are symmetric and Markov chains with these transition matrices have the uniform distribution as a stationary distribution. Lemmas 13 and 14 are on the conductance properties of one-cut and two-cut sets on the c-barbell graph with respect to these transition matrices. In particular, they imply that a set S with minimal conductance should be a one-cut set and has to be given by the vertices of a subgraph $\mathcal{G}_{c_0}$ for some $c_0 \in [1, c]$.
The proof steps of Lemma 14 derives the conductance values of such subgraphs explicitly

$$\Phi_{\mathcal{G}_{c_0}}(\overline{W}_{P^u}) = \frac{1}{c_0}\frac{1}{c\tilde{n}^3}, \quad \Phi_{\mathcal{G}_{c_0}}(\overline{W}_{P^r}) = \frac{1}{2c_0\tilde{n}(c\tilde{n}-1)}. \tag{33}$$

Both of the expressions at (33) are minimized for the choice of $c_0 = \lfloor \frac{c}{2} \rfloor$. Therefore, the minimal conductance is attained for the subgraph $\mathcal{G}_{\lfloor \frac{c}{2} \rfloor}$. Plugging $c_0 = \lfloor \frac{c}{2} \rfloor$ into the expressions above directly yields the graph conductance values at (18). The bounds (19) and (20) follow from Theorem 3 and inequalities (15). ■

Appendix C
Proof of Proposition 9

It follows from Corollary 15 and Lemma 16 that the second largest eigenvalues of $\bar{W}_{P^u}$ and $\bar{W}_{P^r}$ are given by:

$$\begin{aligned} \lambda_{n-1}(\overline{W}_{P^u}) &= 1 - \frac{8}{n^2(n-2)} + \Theta(\frac{1}{n^4}), \\ \lambda_{n-1}(\overline{W}_{P^r}) &= 1 - \frac{1}{n(n-1)} - \Theta(\frac{1}{n^3}). \end{aligned}$$

This implies directly $\lambda_{n-1}(\overline{W}_{P^r}) = 1 - \Theta(\frac{1}{n^2})$ and $\lambda_{n-1}(\overline{W}_{P^u}) = 1 - \Theta(\frac{1}{n^3})$, which completes the proof of Proposition 9.

Appendix D
Supporting Lemmas

The following two lemmas are about the conductance properties of the subsets of a c-barbell graph.

Lemma 13. *Consider a reversible Markov chain on a c-barbell graph with a uniform stationary distribution. Let $\mathcal{H}_0$ be a subgraph of $\mathcal{G}$ whose vertex set is a non-empty two-cut set*

$\mathcal{S}_0$ satisfying $|\mathcal{S}_0| \le \frac{|\mathcal{N}|}{2}$. Then, there exists another subgraph $\tilde{\mathcal{H}}_0$ of $\mathcal{G}$ such that $\Phi_{\tilde{\mathcal{H}}_0}(W) < \Phi_{\mathcal{H}_0}(W)$.

Proof of Lemma 13: Let C_1 and C_2 be the vertex sets of two disjoint non-empty connected subgraphs within $\mathcal{N} \setminus \mathcal{S}_0$ satisfying $\mathcal{N} = C_1 \cup \mathcal{S}_0 \cup C_2$. Note that $C_1 \cap C_2 = \emptyset$ implies either $|C_1 \cup \mathcal{S}_0| \le \frac{|\mathcal{N}|}{2}$ or $|C_2| \le \frac{|\mathcal{N}|}{2}$. Using the fact that the transition matrix W of a reversible Markov chain with a uniform stationary distribution is symmetric, the definition (30) implies $\Phi_{C_1 \cup \mathcal{S}_0}(W) = \Phi_{C_2}(W)$. Without loss of generality, choose $\tilde{\mathcal{H}}_0$ to be the subgraph with vertices $\tilde{\mathcal{S}}_0 = C_1 \cup \mathcal{S}_0$ with $|C_1 \cup \mathcal{S}_0| \le \frac{|\mathcal{N}|}{2}$ (otherwise, pick the subgraph with vertex set C_2 instead), then

$$\begin{aligned}\Phi_{\mathcal{H}_0}(W) &= \frac{1}{|\mathcal{S}_0|}\Big(\sum_{\substack{i\in\mathcal{S}_0\\ j\in C_1}} W_{ij} + \sum_{\substack{i\in\mathcal{S}_0\\ j\in C_2}} W_{ij}\Big)\\ &> \frac{1}{|\mathcal{S}_0|}\sum_{\substack{i\in\mathcal{S}_0\\ j\in C_2}} W_{ij} > \frac{1}{|\tilde{\mathcal{S}}_0|}\sum_{\substack{i\in\tilde{\mathcal{S}}_0\\ j\in C_2}} W_{ij} = \Phi_{\tilde{\mathcal{H}}_0}(W),\end{aligned}$$

which proves Lemma 13. ■

Lemma 14. *Consider a Markov chain on a c-barbell graph with a probability transition matrix W. If $W = \overline{W}_{P^u}$ or $W = \overline{W}_{P^r}$, then for any subgraph $\mathcal{H}_0$ having a one-cut vertex set $\mathcal{S}_0$, there exists a subgraph $\mathcal{G}_{c_0}$ for some $c_0 \in [1, c]$ such that $\Phi_{\mathcal{G}_{c_0}}(W) \le \Phi_{\mathcal{H}_0}(W)$ where $\mathcal{G}_{c_0}$ is defined by* (31) *and* (32).

Proof: For any subgraph $\mathcal{H}_0$ having a one-cut vertex set $\mathcal{S}_0$, we can always a find a subgraph $\mathcal{G}_{c_0}$ with vertex set $\mathcal{V}_{c_0}$ for some $c_0 \in [1, c]$ such that either $\mathcal{V}_{c_0-1} \subset \mathcal{S}_0 \subset \mathcal{V}_{c_0}$ or $\mathcal{V}_{c_0-1} \subset S_0^c \subset \mathcal{V}_{c_0}$ (with the convention that $\mathcal{G}_{c_0}$ is a singleton graph with a vertex set $\mathcal{V}_0$ consisting of a single node). Let $\mathcal{H}_0^c$ be the subgraph with vertex set S_0^c. Since $\Phi_{\mathcal{H}_0}(W) = \Phi_{\mathcal{H}_0^c}(W)$ for both $W = \overline{W}_{P^r}$ and $W = \overline{W}_{P^u}$, without loss of generality we can assume that $\mathcal{H}_0$ satisfies the property $\mathcal{V}_{c_0-1} \subset \mathcal{S}_0 \subset \mathcal{V}_{c_0}$ (otherwise, we can replace $\mathcal{H}_0$ with $\mathcal{H}_0^c$ in the proof below). It follows after a straightforward computation (similar to the proof technique of Lemma 16) that transition probability matrices $\overline{W}_{P^u}$ and $\overline{W}_{P^r}$ on $c - K_{\tilde{n}}$ admit the explicit formula

$$\begin{gathered}[\overline{W}_{P^u}]_{i^*j^*} = \frac{1}{c\tilde{n}^2},\ \ [\overline{W}_{P^u}]_{i^*j} = \frac{1}{2c\tilde{n}^2}\Big(\frac{2\tilde{n}-1}{\tilde{n}-1}\Big),\\ [\overline{W}_{P^u}]_{ij} = \frac{1}{c\tilde{n}(\tilde{n}-1)},\end{gathered}$$

and

$$\begin{gathered}[\overline{W}_{P^r}]_{i^*j^*} = \frac{1}{2(c\tilde{n}-1)},\ \ [\overline{W}_{P^r}]_{i^*j} = \frac{1}{\tilde{n}(c\tilde{n}-1)},\\ [\overline{W}_{P^r}]_{ij} = \frac{1}{\tilde{n}(c\tilde{n}-1)},\end{gathered}$$

where i^* and j^* denote two adjacent nodes belonging to different complete subgraphs of $c - K_{\tilde{n}}$, i.e., those with degree $\tilde{n}$, and $(i, j) \in \mathcal{E}$ or $(i^*, j) \in \mathcal{E}$ such that i and j denote nodes in $c - K_{\tilde{n}}$ with degree $\tilde{n} - 1$. It is worth noting that $[W_{P^r}]_{i^*j^*}$ is greater than $[W_{P^u}]_{i^*j^*}$ as in the $K_{\tilde{n}} - K_{\tilde{n}}$ case. Hence,

$$\Phi_{\mathcal{H}_0}(\overline{W}_{P^u}) = \frac{1}{|\mathcal{S}_0|}\sum_{\substack{i\in\mathcal{S}_0\\ j\in\mathcal{S}_0^c}}[\overline{W}_{P^u}]_{ij} > \frac{1}{c_0\tilde{n}}\frac{1}{c\tilde{n}^2} = \Phi_{\mathcal{G}_{c_0}}(\overline{W}_{P^u}).$$

In the case of $W = \overline{W}_{P^r}$, let $\mathcal{P}_0 \subset \mathcal{S}_0$ be the subset of nodes in the subgraph $K_{\tilde{n}}$ that contains nodes from both $\mathcal{S}_0$ and $\mathcal{S}_0^C$ – if no such $K_{\tilde{n}}$ exists, then $\mathcal{S}_0$ corresponds to a subgraph $\mathcal{G}_{c_0}$ for some $c_0 \in [1, c]$. Now consider the former case, let us denote $m_0 := |\mathcal{P}_0| < \tilde{n}$. The number of edges between $\mathcal{P}_0$ and $\mathcal{S}_0^C$ is given by $m_0(\tilde{n} - m_0)$. This is due to the fact that each node in $\mathcal{P}_0$ has exactly $(\tilde{n} - m_0)$ many edges that connects $\mathcal{S}_0$ to its complement. We have also $m_0(\tilde{n} - m_0) \ge \frac{\tilde{n}}{2}$ for $\tilde{n} \ge 2$. This yields

$$\begin{aligned}\Phi_{\mathcal{H}_0}(\overline{W}_{P^r}) &= \frac{1}{|\mathcal{S}_0|}\sum_{\substack{i\in\mathcal{S}_0\\ j\in\mathcal{S}_0^c}}[\overline{W}_{P^r}]_{ij} \ge \frac{1}{|\mathcal{S}_0|}\frac{m_0(\tilde{n}-m_0)}{\tilde{n}(c\tilde{n}-1)}\\ &\ge \frac{1}{|c_0\tilde{n}|}\frac{1}{2(c\tilde{n}-1)} = \Phi_{\mathcal{G}_{c_0}}(W_{P^r}).\end{aligned}$$

■

Corollary 15. *Under the setting of Proposition 8, assume that the weight matrix w is normalized, i.e., $\sum_{j=1}^n w_{ij} = 1$ for all $i \in \mathcal{N}$. Then $W = w$ is a doubly stochastic matrix and the eigenvalues of W become*

- *$\lambda_a = 1$ with multiplicity one,*
- *$\lambda_b = -1 + (A + G) + F$ with multiplicity one,*
- *$\lambda_c = D - C$ with multiplicity $2\tilde{n} - 4$,*
- *$\lambda_\pm = \frac{1}{2}\Big(F + G - A \pm \sqrt{S}\Big)$,*

where A, B, C, D, E, F, G and S are as in Proposition 8. Moreover, λ_+ satisfies

$$\lambda_+ = \frac{1}{2}\Big(F + G - A + \sqrt{(F - G + A)^2 + 4BE}\Big), \quad (34)$$

and is the second largest eigenvalue, i.e. $\lambda_{n-1}(W) = \lambda_+$.

Proof: Since w is normalized, Proposition 8 applies with $A + G + E = 1$ and $B + F = 1$. Thus eigenvalues simplify to the forms given in the statement. Note that

$$\begin{aligned}\sqrt{S} &= \sqrt{(F + G - A)^2 - 4(FG - BE - AF)}\\ &= \sqrt{(F - G + A)^2 + 4BE} \ge 0.\end{aligned}$$

Therefore, λ_+ satisfies the equality (34). Note that $\lambda_a = 1$ is the unique largest eigenvalue since W is stochastic. It remains to show that λ_+ is the second largest eigenvalue. We can write

$$\begin{aligned}\lambda_+ &= \frac{1}{2}\Big(F + G - A + |F - G + A|\sqrt{1 + \frac{4(n-1)B^2}{(F - G + A)^2}}\Big)\\ &\ge \frac{1}{2}\big(F + G - A + |F - G + A|\big).\end{aligned}$$

There are two cases: $F \ge (G - A)$ or $F < (G - A)$. In both cases, we observe that $\lambda_+ \ge F \ge 0$. Since $A + G + E = 1$, we also have $A + G - 1 = -E \le 0$. Therefore $\lambda_b = F - E \le F \le \lambda_+$. Furthermore, $\lambda_c = D - C \le F = D + (\tilde{n} - 2)C$ since $C \ge 0$; therefore $\lambda_c \le F \le \lambda_+$. Finally, $\lambda_+ \ge 0$ since $S \ge 0$. These observations show that λ_+ is non-negative and is the second largest eigenvalue. ■

Lemma 16. *Consider the setting of Proposition 8:*

(i) *If $W = \overline{W}_{P^u}$, then Proposition 8 applies with $A = A^u$, $B = B^u$, $C = C^u$, $D = D^u$ and $G = G^u$ where*

$$A^u = \frac{2}{n^2},\ B^u = \frac{n-1}{n^2(0.5n-1)},\ C^u = \frac{2}{n(n-2)},$$
$$D^u = \frac{n^3-3n^2+2n+2}{n^2(n-2)},\ G^u = 1-\frac{n+1}{n^2}.$$

The second largest eigenvalue of $\overline{W}_{P^u}$ is given by

$$\begin{aligned}\lambda_{n-1}(\bar{W}_{P^u}) &= 1-\frac{n^2+n-8}{2n^2(n-2)}+\frac{1}{8}\sqrt{S_n^u}\\ &= 1-\frac{8}{n^2(n-2)}+\Theta(\frac{1}{n^4}),\end{aligned}$$

for $S_n^u = \frac{4n^3+24n^2-156n+192}{(0.5n-1)^2n^3}$.

(ii) *If $W = \overline{W}_{P^r}$, then Proposition 8 applies with $A = A^r$, $B = B^r$, $C = C^r$, $D = D^r$ and $G = G^r$ where*

$$A^r = \frac{1}{2(n-1)},\ B^r = \frac{2}{n(n-1)},\ C^r = \frac{2}{n(n-1)},$$
$$D^r = \frac{n^2-2n+2}{n(n-1)},\ G^r = 1-\frac{1.5n-2}{n(n-1)}.$$

Moreover, the second largest eigenvalue of $\overline{W}_{P^u}$ is given by

$$\begin{aligned}\lambda_{n-1}(\bar{W}_{P^r}) &= 1-\frac{1}{(n-1)}+\frac{1}{2}\sqrt{S_n^r}\\ &= 1-\frac{1}{n(n-1)}-\Theta(\frac{1}{n^3}),\end{aligned}$$

for $S_n^r = \frac{4n-8}{n(n-1)^2}$.

Proof: We first compute the entries of both P^u and P^r matrices explicitly for the barbell graph (i.e. $K_{\tilde{n}}-K_{\tilde{n}}$). Former one can be found directly from degrees of the nodes,

$$P_{ij}^u = \begin{cases}\frac{1}{2\tilde{n}(\tilde{n}-1)} & \text{if } i \notin \{i^*, j^*\},\\ \frac{1}{2\tilde{n}^2} & \text{if } i \in \{i^*, j^*\}.\end{cases}$$

Calculating P^r requires us to find effective resistances on the graph. Following definition of resistance allows us to calculate them using Cayley's formula for complete graphs,

$$R_{ij} = \frac{\text{\# of spanning trees passing through } (i,j)}{\text{\# of spanning trees}}.$$

Remember that complete graph with $\tilde{n}$ vertices has $\tilde{n}^{\tilde{n}-2}$ spanning trees, therefore barbell graph has $\tilde{n}^{2\tilde{n}-4}(\tilde{n}^{\tilde{n}-2} \times \tilde{n}^{\tilde{n}-2})$ spanning trees. Let K be the number of trees passing from an edge then $K \times \binom{\tilde{n}}{2} = \tilde{n}^{\tilde{n}-2}(\tilde{n}-1)$. So we have $K = 2\tilde{n}^{\tilde{n}-3}$. This implies that number of spanning trees passing from an edge is $2\tilde{n}^{2\tilde{n}-5}$ on barbell graph, and definitely the number of spanning trees passing from the edge (i^*, j^*) is $\tilde{n}^{2\tilde{n}-4}$. This implies, $R_{ij} = 1$ if $(i,j) \in \{(i^*, j^*), (j^*, i^*)\}$, $R_{ij} = \frac{2}{\tilde{n}}$ otherwise. Once we have explicit characterizations of P^u and P^r, using Lemma 2 we can compute the entries of $\overline{W}_{P^u}$ and $\overline{W}_{P^r}$ to be given as in (i) and (ii). The second largest eigenvalues of matrices $\bar{W}_{P^u}$ and $\bar{W}_{P^r}$ follow from Corollary 15. ■

Lemma 17. *[65, Eqn. (2.2)] Let W be the transition matrix of a Markov chain with stationary distribution π. Let j be a neighbor of i, i.e. $j \in \mathcal{N}_i$, then $H_W(i \to j) \leq (\pi_j W_{ji})^{-1}$.*

References

[1] N. S. Aybat and M. Gürbüzbalaban, "Decentralized computation of effective resistances and acceleration of consensus algorithms," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 538–542.

[2] D. J. Klein, "Resistance-distance sum rules," *Croatica chemica acta*, vol. 75, no. 2, pp. 633–649, 2002.

[3] D. J. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF01164627

[4] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM review*, vol. 50, no. 1, pp. 37–66, 2008.

[5] D. Aldous and J. A. Fill, "Reversible Markov chains and random walks on graphs," 2014, unfinished monograph, available at: http://www.stat.berkeley.edu/∼aldous/RWG/book.html.

[6] P. G. Doyle and J. L. Snell, *Random walks and electric networks*. Mathematical Association of America,, 1984.

[7] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: Spectral properties and applications," in *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006, pp. 3479–3485.

[8] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011. [Online]. Available: http://dx.doi.org/10.1137/080734029

[9] N. Anari and S. O. Gharan, "Effective-resistance-reducing flows, spectrally thin trees, and asymmetric tsp," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, Oct 2015, pp. 20–39.

[10] A. Tizghadam and A. Leon-Garcia, "Betweenness centrality and resistance distance in communication networks," *IEEE Network*, vol. 24, no. 6, pp. 10–16, November 2010.

[11] A. Jadbabaie, "On geographic routing without location information," in *43rd IEEE Conference on Decision and Control (CDC), 2004.*, vol. 5. IEEE, 2004, pp. 4764–4769.

[12] M. Kapralov and R. Panigrahy, "Spectral sparsification via random spanners," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 393–398.

[13] A. Pirotte, J.-M. Renders, M. Saerens *et al.*, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge & Data Engineering*, no. 3, pp. 355–369, 2007.

[14] R. B. Bapat, I. Gutmana, and W. Xiao, "A simple method for computing resistance distance," *Zeitschrift für Naturforschung A*, vol. 58, no. 9-10, pp. 494–498, 2003.

[15] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2508–2530, 2006.

[17] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[18] A. Ene and A. Vladu, "Improved convergence for ℓ_1 and ℓ regression via iteratively reweighted least squares," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 1794–1801. [Online]. Available: http://proceedings.mlr.press/v97/ene19a.html

[19] S. Park, D. Deka, and M. Chertkov, "Learning in power distribution grids under correlated injections," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018, pp. 1863–1868.

[20] X. Wang, E. Pournaras, R. E. Kooij, and P. Van Mieghem, "Improving robustness of complex networks via the effective graph resistance," *The European Physical Journal B*, vol. 87, no. 9, p. 221, Sep 2014.

[21] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng, "Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs," in *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, ser. STOC '11. New York, NY, USA: ACM, 2011, pp. 273–282. [Online]. Available: http://doi.acm.org/10.1145/1993636.1993674

[22] W. S. Rossi, P. Frasca, and F. Fagnani, "Average resistance of toroidal graphs," *SIAM journal on control and optimization*, vol. 53, no. 4, pp. 2541–2557, 2015.

[23] V. L. Alev, N. Anari, L. C. Lau, and S. O. Gharan, "Graph clustering using effective resistance," *CoRR*, vol. abs/1711.06530, 2017. [Online]. Available: http://arxiv.org/abs/1711.06530

[24] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[25] Y. Zhang and X. Lin, "DISCO: Distributed optimization for self-concordant empirical loss," in *International conference on machine learning*, 2015, pp. 362–370.

[26] C.-p. Lee, C. H. Lim, and S. J. Wright, "A distributed quasi-newton algorithm for empirical risk minimization with nonsmooth regularization," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1646–1655.

[27] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2018.

[28] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2016.

[29] N. Loizou and P. Richtárik, "Accelerated gossip via stochastic Heavy Ball method," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 927–934.

[30] ——, "Revisiting randomized gossip algorithms: General framework, convergence rates and novel block and accelerated protocols," *arXiv preprint arXiv:1905.08645*, 2019.

[31] N. Loizou, M. Rabbat, and P. Richtárik, "Provably accelerated randomized gossip algorithms," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7505–7509.

[32] P. Rebeschini and S. C. Tatikonda, "Accelerated consensus via min-sum splitting," in *Advances in Neural Information Processing Systems*, 2017, pp. 1374–1384.

[33] R. L. G. Cavalcante, A. Rogers, and N. R. Jennings, "Consensus acceleration in multi-agent systems with the Chebyshev semi-iterative method," in *The 10th International Conference on Autonomous Agents and Multi-agent Systems - Volume 1*, ser. AAMAS '11. Richland, SC: International Foundation for Autonomous Agents and Multi-agent Systems, 2011, pp. 165–172. [Online]. Available: http://dl.acm.org/citation.cfm?id=2030470.2030494

[34] J. H. Seidman, M. Fazlyab, G. J. Pappas, and V. M. Preciado, "A Chebyshev-accelerated primal-dual method for distributed optimization," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1775–1781.

[35] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij, "Effective graph resistance," *Linear algebra and its applications*, vol. 435, no. 10, pp. 2491–2506, 2011.

[36] E. Y. Hamedani and N. S. Aybat, "Multi-agent constrained optimization of a strongly convex function," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 558–562.

[37] ——, "Multi-agent constrained optimization of a strongly convex function over time-varying directed networks," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 518–525.

[38] R. M. Gower and P. Richtárik, "Stochastic dual ascent for solving linear systems," *arXiv preprint arXiv:1512.06890*, 2015.

[39] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2009.

[40] A. Zouzias and N. M. Freris, "Randomized extended Kaczmarz for solving least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 2, pp. 773–793, 2013.

[41] P. Tetali, "An extension of Foster's network theorem," *Combinatorics, Probability and Computing*, vol. 3, no. 3, pp. 421–427, 1994.

[42] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012, vol. 3.

[43] K. Jung, D. Shah, and J. Shin, "Distributed averaging via lifted Markov chains," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 634–647, 2010.

[44] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Symmetry analysis of reversible Markov chains," *Internet Mathematics*, vol. 2, no. 1, pp. 31–71, 2005. [Online]. Available: https://doi.org/10.1080/15427951.2005.10129100

[45] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of Markov chains," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 36–61, 1991. [Online]. Available: http://www.jstor.org/stable/2959624

[46] K. Censor-Hillel and H. Shachnai, "Fast information spreading in graphs with large weak conductance," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1451–1465, 2012.

[47] D. Shah, "Gossip algorithms," *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2009.

[48] Y. Peres and P. Sousi, "Mixing times are hitting times of large sets," *Journal of Theoretical Probability*, vol. 28, no. 2, pp. 488–519, 2015.

[49] A. Olshevsky, "Linear time average consensus on fixed graphs?" *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 94–99, 2015.

[50] S. Roch *et al.*, "Bounding fastest mixing," *Electronic Communications in Probability*, vol. 10, pp. 282–296, 2005.

[51] R. Cont and E. Tanimura, "Small-world graphs: Characterization and alternative constructions," *Advances in Applied Probability*, vol. 40, no. 4, pp. 939–965, 2008. [Online]. Available: http://www.jstor.org/stable/20443619

[52] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari, "The electrical resistance of a graph captures its commute and cover times," *Computational Complexity*, vol. 6, no. 4, pp. 312–340, 1996.

[53] D. A. Levin, Y. Peres, and E. L. Wilmer, "Markov chains and mixing times. with a chapter by James G. Propp and David B. Wilson," *American Mathematical Society, Providence, RI*, 2009.

[54] N. K. Vishnoi *et al.*, "Lx= b," *Foundations and Trends® in Theoretical Computer Science*, vol. 8, no. 1–2, pp. 1–141, 2013.

[55] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.

[56] E. Zanaj, M. Baldi, and F. Chiaraluce, "Efficiency of the gossip algorithm for wireless sensor networks," in *2007 15th International Conference on Software, Telecommunications and Computer Networks*, Sep. 2007, pp. 1–5.

[57] A.-M. Kermarrec and M. van Steen, "Gossiping in distributed systems," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 2–7, Oct. 2007. [Online]. Available: http://doi.acm.org/10.1145/1317379.1317381

[58] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 482–491.

[59] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 634–649, May 2008.

[60] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 263–270. [Online]. Available: http://doi.acm.org/10.1145/313451.313556

[61] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[62] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, Aug. 2012.

[63] Mosek ApS, "The Mosek optimization toolbox for Matlab manual. version 8.0.0.60," http://docs.mosek.com/8.0/toolbox/index.html, 2019.

[64] F. Zhang, *The Schur Complement and Its Applications*, ser. Numerical Methods and Algorithms. Springer US, 2006. [Online]. Available: https://books.google.com/books?id=EMEyg8NcuskC

[65] S. Ikeda, I. Kubo, and M. Yamashita, "The hitting and cover times of random walks on finite graphs using local degree information," *Theoretical Computer Science*, vol. 410, no. 1, pp. 94–100, 2009.

Supplementary File

PROOF OF PROPOSITION 8

The proof follows by adapting the proof of [44, Proposition 5.1] to our setting with minor modifications. It is based on exploiting the symmetry group properties of the barbell graph with algebraic techniques. We first give relevant background material below before going into the details of the proof.

Background Material

Consider a weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, w)$. A *permutation* $p : \mathcal{N} \to \mathcal{N}$ is a mapping that rearranges the vertices, i.e. it is a bijection from the node set $\mathcal{N}$ to itself. We consider a *permutation group* H, which is a group whose elements are permutations of $\mathcal{N}$ and whose group operation is the composition of permutations in H. By the group property, if two permutations $s_1, s_2 \in H$, then the composition $s_1 s_2 \in H$ and in particular the *identity permutation* e which maps all the elements of $\mathcal{N}$ to itself is also contained in H. The group that contains all the $n!$ permutations defined on $\mathcal{N}$ is denoted as S_n.

The *direct product* $(H_1 \times H_2)$ of two groups H_1, H_2 is defined as the group that consists of elements from the Cartesian product of H_1 and H_2 with the elementwise composition, i.e. $(h_1, h_2) \in (H_1 \times H_2)$ if and only if $h_1 \in H_1$ and $h_2 \in H_2$ and if $(h_1, h_2) \in (H_1 \times H_2)$ and $(\tilde{h}_1, \tilde{h}_2) \in (\tilde{H}_1 \times \tilde{H}_2)$ then the composition operation $\cdot$ over $(H_1 \times H_2)$ is defined as $(h_1, h_2) \cdot (\tilde{h}_1, \tilde{h}_2) = (h_1 \tilde{h}_1, h_2 \tilde{h}_2)$. A subgroup M of a group H is *normal* if for all $h \in H$ and $m \in M$ we have $hmh^{-1} \in M$. The *semidirect product* $H_1 \ltimes H_2$ of two groups H_1 and H_2 is the group that consists of elements $h = h_1 h_2$ with $h_1 \in H_1$ and $h_2 \in H_2$ and the subgroup H_1 is normal in $H_1 \ltimes H_2$ with the condition $H_1 \cap H_2 = \{e\}$. The *orbit* O_i of an element $i \in \mathcal{N}$, under a permutation group H is the set $O_i := \{v \in \mathcal{N} \mid \exists s \in H \text{ s.t. } s(v) = i\}$. In other words, the orbit of node i is the set of vertices that can be mapped to i by an element of the permutation group H. This definition creates an equivalence relation $\sim$ on $\mathcal{N}$; for $i, j \in \mathcal{N}$, we say $i \sim j$ if $O_i = O_j$. In particular, equivalence classes form a partition of $\mathcal{N}$.

A permutation s is called *an automorphism* of the weighted graph $\mathcal{G}$ if the weight matrix w is invariant under s, i.e. if $w(i,j) = w(s(i), s(j))$. From this definition, an automorphishm s also satisfies $W(i,j) = W(s(i), s(j))$ where $W(i,j) = w(i,j)/\sum_{j \in \mathcal{N}_i} w(i,j)$ is the transition probability. We are interested in such permutations that preserve the structure of w and therefore W. The group of all automorphisms with the operation of composition of permutations is called the *automorphism group* of the graph and is denoted by $\text{Aut}(\mathcal{G})$. Let S be a subgroup of $\text{Aut}(\mathcal{G})$ and consider the orbits $\{O_i\}_{i \in \mathcal{N}}$ under the permutation group S which partition the set $\mathcal{N}$. We define *orbit graph* to be the graph whose vertices consist of the equivalence classes O_i for $i \in \mathcal{N}$ and we consider an induced Markov chain on the orbit graph with probability transition probabilities defined as

$$W_S(O_i, O_j) = \sum_{j' \in O_j} W(i, j'). \tag{35}$$

This Markov chain is also called the *orbit chain*. It can be shown that the definition of the weights W_S above does not depend on the choice of the element i from the set O_i (see e.g. [44]).

Proof

First, we consider the automorphism group of the barbell graph $K_{\tilde{n}} - K_{\tilde{n}}$ with edge weights given by Proposition 8. Consider the nodes i_* and j_* that connect the complete subgraphs of the barbell graph and without loss of generality assume that we enumerate the nodes so that $i_* = \tilde{n}$, $j_* = \tilde{n} + 1$ and a node $i < \tilde{n}$ is on the complete subgraph on the left hand-side and any node $j > \tilde{n} + 1$ is on the complete subgraph on the right-hand side. We see from the symmetry structure of W that if we take any two nodes from a complete subgraph and permute them, this would be an automorphism. Similarly, swapping the two complete subgraphs between them would be an automorphism; i.e. the permutation $C_2 : \mathcal{N} \to \mathcal{N}$ that maps $i \overset{C_2}{\mapsto} -i \mod (n+1)$ is an automorphism. It follows from these observations that the automorphism group of $K_{\tilde{n}} - K_{\tilde{n}}$ is the group $C_2 \ltimes (S_{\tilde{n}-1} \times S_{\tilde{n}-1})$ (see also [44] for more details). It is known that for any subgroup S of the automorphism group, the eigenvalues of the transition matrix W_S defined by (35) should also be an eigenvalue of the transition matrix W (see e.g. [44, Section 3]). Note that the square matrix W_S has dimension $n_S \times n_S$ where $n_S \leq n$, so the set of eigenvalues of W_S are a subset of the set of all eigenvalues of W. We are going to use this result to prove the Proposition 8. Next, we consider the eigenvalues of the transition matrices W_S of the orbit chains under subgroups S of $C_2 \ltimes (S_{\tilde{n}-1} \times S_{\tilde{n}-1})$:

a) The orbit chain under $C_2 \ltimes (S_{\tilde{n}-1} \times S_{\tilde{n}-1})$ (Figure 8) has the transition matrix

$$\begin{bmatrix} \frac{A+G}{A+G+E} & \frac{E}{A+G+E} \\ \frac{E}{(n-1)F+E} & \frac{(n-1)F}{(n-1)F+E} \end{bmatrix}.$$

Since $\lambda_a = 1$ is an eigenvalue, and its trace is the sum of eigenvalues; it follows that the other eigenvalue of this matrix is given by $\lambda_b = -1 + \frac{A+G}{A+G+E} + \frac{F}{F+B}$.

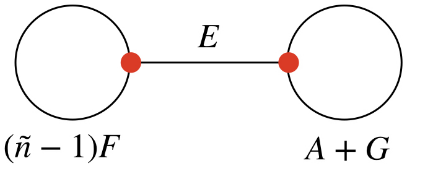

Fig. 8: Orbit graph under $C_2 \ltimes (S_{\tilde{n}-1} \times S_{\tilde{n}-1})$

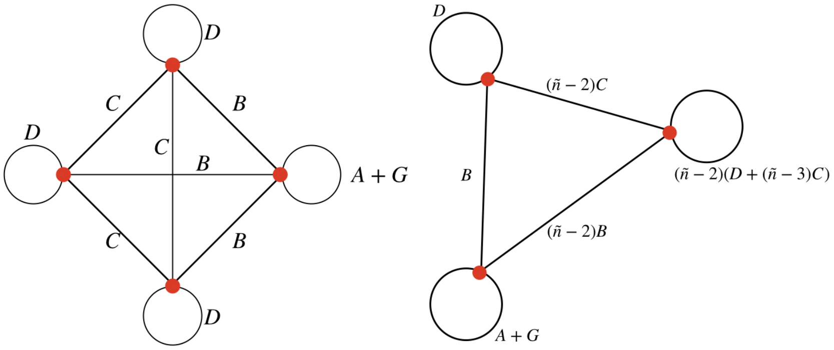

Fig. 9: **Left:** Orbit graph under C_2. **Right:** Orbit graph under $C_2 \ltimes (S_{\tilde{n}-2} \times S_{\tilde{n}-2})$

b) Consider the orbit chain under C_2 illustrated on the left panel of Figure 9. This orbit graph has two orbits under permutation $S_{\tilde{n}-1}$: One of them contains only one node (the node with a self-loop with weight $(A+G)$) and the other orbit has the remaining $\tilde{n}-1$ nodes. Notice that the latter orbit has identical $\tilde{n}-1$ elements and therefore the permutation group $C_2 \ltimes (S_{\tilde{n}-2} \times S_{\tilde{n}-2})$ fixes one of the nodes having a loop with weight D and permutes the remaining $\tilde{n}-2$ nodes among themselves without affecting the orbit with one node. Therefore, by [44, Thereom 3.1], the eigenvalues of the transition matrix W' of the orbit graph obtained by the permutation group $S = C_2 \ltimes (S_{\tilde{n}-2} \times S_{\tilde{n}-2})$ (illustrated on the right panel of Figure 9) are also eigenvalues of the transition matrix W. The transition matrix W' is 3×3 with three eigenvalues, including λ_a and λ_b that we have already found at part (a). The third eigenvalue λ_c can be computed from the transition matrix W' of the orbit chain under $C_2 \ltimes (S_{\tilde{n}-2} \times S_{\tilde{n}-2})$:

$$\begin{bmatrix} \frac{A+G}{(\tilde{n}-1)B+A+G} & * & * \\ * & \frac{D}{(\tilde{n}-2)C+D+B} & * \\ * & * & \frac{D+(\tilde{n}-3)C}{B+D+(\tilde{n}-2)C} \end{bmatrix},$$

where we use $*$ to denote the entries of this matrix that will not be relevant to our discussion. In particular, the eigenvalues of this matrix will be λ_a, λ_b and λ_c; the latter will be an eigenvalue of W with multiplicity $2\tilde{n}-4$. Again, using the fact that the trace of a matrix is equal to the sum of its eigenvalues, we obtain

$$\lambda_c = \frac{D-C}{F+B}.$$

c) Lastly, orbit chain under $(S_{\tilde{n}-1} \times S_{\tilde{n}-1})$ consists of four orbits: $(\tilde{n}-1)$ points in the left and right complete graphs and vertices i_* and j_* as illustrated in Figure 10.

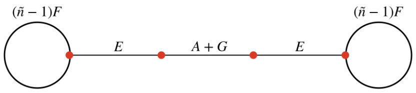

Fig. 10: Orbit graph under $S_{\tilde{n}-1} \times S_{\tilde{n}-1}$

This orbit chain has the transition matrix of the form

$$\begin{bmatrix} \frac{F}{B+F} & \frac{B}{B+F} & 0 & 0 \\ \frac{E}{A+E+G} & \frac{G}{A+E+G} & \frac{A}{A+E+G} & 0 \\ 0 & \frac{A}{A+E+G} & \frac{G}{A+E+G} & \frac{E}{A+E+G} \\ 0 & 0 & \frac{B}{B+F} & \frac{F}{B+F} \end{bmatrix}.$$

After a straightforward computation, it can be checked that this matrix has the eigenvalues, $1, \lambda_+, \lambda_-, (-1+\frac{A+G}{A+E+G}+\frac{F}{B+F})$ where

$$\lambda_\pm = \frac{1}{2}\left[\frac{F}{B+F}+\frac{G-A}{A+E+G}\pm\sqrt{S}\right],$$

and $S=\left(\frac{F}{B+F}+\frac{G-A}{A+E+G}\right)^2-\frac{4(FG-BE-AF)}{(B+F)(A+E+G)}$.

Remark 18. *Boyd* et al. *[44] studied the case $W_{i^*i^*}=0=W_{j^*j^*}$ where similar orbit chains and graphs arise. The proof of Proposition 8 given here is a minor modification of the original proof of Boyd et al. [44, Proposition 2.2] and extends it to the more general case where $W_{i^*i^*}$ or $W_{j^*j^*}$ can be strictly positive.*