Internet Mathematics, 11:449–471, 2015
Copyright © Taylor & Francis Group, LLC
ISSN: 1542-7951 print/1944-9488 online
DOI: 10.1080/15427951.2015.1009522

SOLVING LOCAL LINEAR SYSTEMS WITH BOUNDARY CONDITIONS USING HEAT KERNEL PAGERANK

Fan Chung and Olivia Simpson
Department of Computer Science and Engineering, University of California, San Diego, California, USA

Abstract *We present an efficient algorithm for solving local linear systems with a boundary condition using the Green's function of a connected induced subgraph related to the system. We introduce the method of using the Dirichlet heat kernel pagerank*[1] *vector to approximate local solutions to linear systems in the graph Laplacian, satisfying given boundary conditions over a particular subset of vertices. With an efficient algorithm for approximating Dirichlet heat kernel pagerank, our Local Linear Solver algorithm computes an approximate local solution with multiplicative and additive error* ϵ *by performing* $O(\epsilon^{-5}s^3 \log(s^3\epsilon^{-1}) \log n)$ *random walk steps, where* n *is the number of vertices in the full graph, and* s *is the size of the local system on the induced subgraph.*

1. INTRODUCTION

There are a number of linear systems that model flow over vertices of a graph with a given boundary condition. A classical example is the case of an electrical network. Flow can be captured by measuring electric current between points in the network and the amount that is injected and removed from the system. Here, the points at which voltage potential is measured can be represented by vertices in a graph, and edges are associated to the ease with which current passes between two points. The injection and extraction points can be viewed as the boundary of the system, and the relationship of the flow and voltage can be evaluated by solving a system of linear equations over the measurement points.

Another example is a decision-making process among a network of agents. Each agent decides on a value, but may be influenced by the decision of other agents in the network. Over time, the goal is to reach consensus among all the agents, in which each agrees on a common value. Agents are represented by vertices, and each vertex has an associated value. The amount of influence an agent has on a fellow agent is modeled by a weighted edge between the two representative vertices, and the communication dynamics can be modeled by a linear system. In this case, some special agents, which make their own decisions, can be viewed as the boundary.

[1] Heat kernel pagerank is a variation of an algorithm known as Personalized PageRank [7]. PageRank is used by Google to rank websites as a way of measuring the importance of website pages.

Address correspondence to Olivia Simpson, Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-mail: osimpson@ucsd.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uinm.

In both these cases, the linear systems are equations formulated in the graph Laplacian. Spectral properties of the Laplacian are closely related to reachability and the rate of diffusion across vertices in a graph [4]. Laplacian systems have been used to concisely characterize qualities such as edge resistance and the influence of communication on edges [23]. There is a substantial body of work on efficient and nearly linear time solvers for Laplacian linear systems [2, 8, 10, 13–17, 21, 22, 24, 26]; see also [27].

The focus of this article is a localized version of a Laplacian linear solver. In a large network, possibly of hundreds of millions of vertices, the algorithms we are dealing with and the solutions we are seeking are usually of finite support. Here, by finite we mean the support size depends only on the requested output and is independent of the full size of the network. Sometimes we allow sizes up to a factor of $\log(n)$, where n is the size of the network.

The setup is a graph and a boundary condition given by a vector with specified limited support over the vertices. In the local setting, rather than computing the full solution, we compute the solution over a fraction of the graph and de facto ignore the vertices with solution values below the multiplicative/additive error bound. In essence, we avoid computing the entire solution by focusing computation on the subset itself. In this way, computation depends on the size of the subset, rather than the size of the full graph. We distinguish the two cases as "global" and "local" linear solvers, respectively. We remark that in the case the solution is not "local," for example, if *all* values are below the error bound, our alogrithm will return the zero vector—a valid approximate solution according to our definition of approximation.

In this article, we show how local Laplacian linear systems with a boundary condition can be solved and efficiently approximated by using *Dirichlet heat kernel pagerank*, a diffusion process over an induced subgraph. We will illustrate the connection between the Dirichlet heat kernel pagerank vector and the *Green's function*, or the inverse of a submatrix of the Laplacian determined by the subset. We also demonstrate the method of approximation using random walks. Our algorithm approximates the solution to the system restricted to the subset S by performing $O(\gamma^{-2}\epsilon^{-3}s^3 \log^2(s^3\gamma^{-1}) \log n)$ random walk steps, where γ is the error bound for the solver and ϵ is the error bound for Dirichlet heat kernel pagerank approximation, and s denotes the size of S. We assume that performing a random walk step and drawing from a distribution with finite support requires constant time. With this, our algorithm runs in time $O(\gamma^{-2}\epsilon^{-3} \log^4(n) \log^2(\gamma^{-1} \log^3(n)))$ when the support size of the solution is $O(\log n)$. Note that in our computation, we do not intend to compute or approximate the matrix form of the inverse of the Laplacian. We intend to compute an approximate local solution, which is optimal subject to the (relaxed) definition of approximation.

1.1. A summary of the main results

We give an algorithm called *Local Linear Solver* for approximating a local solution of a Laplacian linear system with a boundary condition. The algorithm uses the connection between the inverse of the restricted Laplacian and the Dirichlet heat kernel of the graph for approximating the local solution with a sampling of Dirichlet heat kernel pagerank vectors (heat kernel pagerank restricted to a subset S). It is shown in Theorem 4.6 that the output of *Local Linear Solver* approximates the exact local solution x_S with absolute error $O(\gamma \|b\| + \|x_S\|)$ for boundary vector b with probability at least $1 - \gamma$.

We present an efficient algorithm for approximating Dirichlet heat kernel pagerank vectors, `ApproxDirHKPR`. The algorithm is an extension of the algorithm in [7]. The definition of ϵ-approximate vectors is given in Section 5. We note that this notion of approximation is weaker than the classical notions of total variation distance among others. Nevertheless, this "relaxed" notion of approximation is used in analyzing PageRank algorithms (see [3], for example) for massive networks.

The full algorithm for approximating a local linear solution, `GreensSolver`, is presented in Section 6. The algorithm is an invocation of *Local Linear Solver* with the `ApproxDirHKPR` called as a subroutine. The full agorithm requires $O(\gamma^{-2}\epsilon^{-3}s^3\log^2(s^3\gamma^{-1})\log n)$ random walk steps by using the algorithm `ApproxDirHKPR` with a slight modification. Our algorithm achieves sublinear time after preprocessing, which depends on the size of the support of the boundary condition. The error is similar to the error of `ApproxDirHKPR`.

It is worth pointing out a number of ways our methods can be generalized. First, we focus on unweighted graphs, though extending our results to graphs with edge weights follows easily with a weighted version of the Laplacian. Second, we require the induced subgraph on the subset S be connected. However, if the induced subgraph is not connected, the results can be applied to components separately, so our requirement on connectivity can be relaxed. Finally, we restrict our discussion to linear systems in the graph Laplacian. However, by using a linear-time transformation attributable to [11] for converting a symmetric, diagonally dominant linear system to a Laplacian linear system, our results apply to a larger class of linear systems.

1.2. Organization

In Section 2, we give definitions and basic facts for graph Laplacian and heat kernel. In Section 3 the problem is introduced in detail and provides the setting for the local solver. The algorithm, *Local Linear Solver*, is presented in Section 4. After this, we extend the solver to the full approximation algorithm using approximate Dirichlet heat kernel pagerank. In Section 5, we give the definition of local approximation and analyze the Dirichlet heat kernel pagerank approximation algorithm. In Section 6, the full algorithm for computing an approximate local solution to a Laplacian linear system with a boundary condition, `GreensSolver`, is given. Finally, in Section 7 we illustrate the correctness of the algorithm with an example network and specified boundary condition. The example demonstrates visually what a local solution is and how `GreensSolver` successfully approximates the solution within the prescribed error bounds when the solution is sufficiently local.

2. BASIC DEFINITIONS AND FACTS

Let G be a simple graph given by vertex set $V = V(G)$ and edge set $E = E(G)$. Let $u \sim v$ denote $\{u, v\} \in E$. When considering a real vector f defined over the vertices of G, we say $f \in \mathbb{R}^V$ and the *support* of f is denoted by $\mathrm{supp}(f) = \{v \in V : f(v) \neq 0\}$. For a subset of vertices $S \subseteq V$, we say $s = |S|$ is the size of S and use $f \in \mathbb{R}^S$ to denote vectors defined over S. When considering a real matrix M defined over V, we say $M \in \mathbb{R}^{V\times V}$, and we use M_S to denote the submatrix of M with rows and columns indexed by vertices in S. Namely, $M_S \in \mathbb{R}^{S\times S}$. Similarly, for a vector $f \in \mathbb{R}^V$, we use f_S to mean the subvector of f with entries indexed by vertices in S. The *vertex boundary*

of S is $\delta(S) = \{u \in V \setminus S : \{u, v\} \in E \text{ for some } v \in S\}$, and the *edge boundary* is $\partial(S) = \{\{u, v\} \in E : u \in S, v \notin S\}$.

2.1. Graph Laplacians and heat kernel

For a graph G, let A be the indicator adjacency matrix $A \in \{0, 1\}^{V \times V}$ for which $A_{uv} = 1$ if and only if $\{u, v\} \in E$. The *degree* of a vertex v is the number of vertices adjacent to it, $d_v = |\{u \in V | A_{uv} = 1\}|$. Let D be the diagonal degree matrix with entries $D_{vv} = d_v$ on the diagonal and zero entries elsewhere. The *Laplacian* of a graph is defined to be $L = D - A$. The *normalized Laplacian*, $\mathcal{L} = D^{-1/2} L D^{-1/2}$, is a degree-nomalized formulation of L, given by

$$\mathcal{L}(u, v) = \begin{cases} 1 & \text{if } u = v, \\ \frac{-1}{\sqrt{d_u d_v}} & \text{if } u \sim v, \\ 0 & \text{otherwise.} \end{cases}$$

Let $P = D^{-1} A$ be the transition probability matrix for a random walk on the graph. Namely, if v is a neighbor of u, then $P(u, v) = 1/d_u$ denotes the probability of moving from vertex u to vertex v in a random walk step. Another related matrix of significance is the *Laplace operator*, $\Delta = I - P$. We note that $\mathcal{L}$ is similar to Δ.

The *heat kernel* of a graph is defined for real $t > 0$ by

$$\mathcal{H}_t = e^{-t\mathcal{L}}.$$

Consider a similar matrix, denoted by $H_t = e^{-t\Delta} = D^{-1/2} \mathcal{H}_t D^{1/2}$. For a given $t \in \mathbb{R}^+$ and a preference vector $f \in \mathbb{R}^V$, the *heat kernel pagerank* is defined by

$$\rho_{t,f} = f^T H_t,$$

where f^T denotes the transpose of f. When f is a probability distribution on V, we can also express the heat kernel pagerank as an exponential sum of random walks. Here we follow the notation for random walks so that a random walk step is by a right multiplication by P:

$$\rho_{t,f} = f^T e^{-t\Delta} = e^{-t} \sum_{k=0}^{\infty} \frac{t^k}{k!} f^T P^k.$$

2.2. Laplacian linear system

The examples of computing current flow in an electrical network and consensus in a network of agents typically require solving linear systems with a boundary condition formulated in the Laplacian $L = D - A$, where D is the diagonal matrix of vertex degrees and A is the adjacency matrix of the network. The problem in the global setting is the solution to $L\mathtt{x} = \mathtt{b}$, whereas the solution $\mathtt{x}$ is required to satisfy the boundary condition $\mathtt{b}$ in the sense that $\mathtt{x}(v) = \mathtt{b}(v)$ for every vertex v in the support of $\mathtt{b}$. Because our analysis uses random walks, we use the normalized Laplacian $\mathcal{L} = D^{-1/2} L D^{-1/2}$. We note that the solution x for Laplacian linear equations of the form $\mathcal{L}x = b$ is equivalent to solving $L\mathtt{x} = \mathtt{b}$ if we take $\mathtt{x} = D^{-1/2}x$ and $\mathtt{b} = D^{1/2}b$. Specifically, our local solver computes

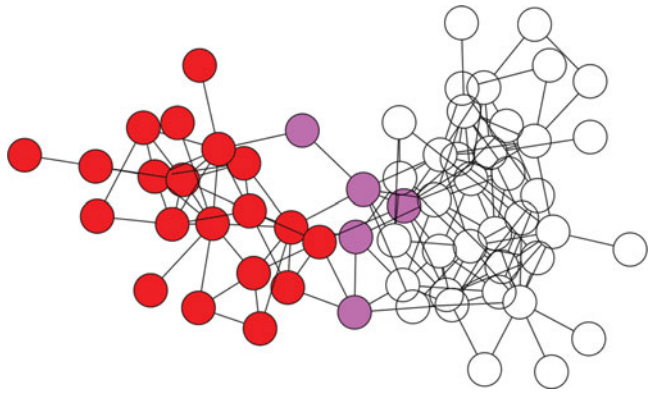

Figure 1 A communication network of agents in which the leaders (shaded black) have fixed decisions and the followers (shaded grey) compute their decisions based on the leaders and subnetwork of followers. The local solution would be the decisions of the followers.

the solution x restricted to S, denoted x_S, and we do this by way of the discrete Green's function.

2.2.1. Example. To illustrate the local setting, we expand on the problem of a network of decision-making agents. Consider a communication network of agents in which a certain subset of agents $f \subset V$ are *followers* and an adjacent subset $l \subset V \setminus f$ are *leaders* (see Figure 1). Imagine that the decision values of each agent depend on neighbors as usual, but also that the values of the leaders are fixed and will not change. Specifically, let d_v denote the degree of agent v, or the number of adjacent agents in the communication network, and let x be a vector of decision values of the agents. Suppose every follower v_f continuously adjusts its decision according to the protocol:

$$x(v_f) = x(v_f) - \frac{1}{\sqrt{d_{v_f}}} \sum_{u \sim v_f} \frac{x(u)}{\sqrt{d_u}},$$

while every leader v_l remains fixed at $b(v_l)$. Then the vector of decision values x is the solution to the system $\mathcal{L}x = b$, where x is required to satisfy the boundary condition.

In our example, we are interested in computing the decision values of the followers of the network in which the values of the leaders are a fixed boundary condition but continue to influence the decisions of the subnetwork of followers.

3. SOLVING LOCAL LAPLACIAN LINEAR SYSTEMS WITH A BOUNDARY CONDITION

For a general connected, simple graph G and a subset of vertices S, consider the linear system $\mathcal{L}x = b$, where the vector b has nonempty support on the vertex boundary of S. The global problem is finding a solution x that agrees with b, in the sense that $x(v) = b(v)$ for every vertex b in the support of b. In this case we say that x *satisfies the boundary condition* b.

Specifically, for a vector $b \in \mathbb{R}^V$, let S denote a subset of vertices in the complement of $\mathrm{supp}(b)$. Then b can be viewed as a function defined on the vertex boundary $\delta(S)$ of S

and we say b is a *boundary condition* of S. Here we will consider the case that the induced subgraph on S is connected.

Definition 3.1. Let G be a graph and let b be a vector $b \in \mathbb{R}^V$ over the vertices of G with nonempty support. Then we say a subset of vertices $S \subset V$ is a *b-boundable subset* if

(i) $S \subseteq V \setminus \mathrm{supp}(b)$,
(ii) $\delta(S) \cap \mathrm{supp}(b) \neq \emptyset$,
(iii) the induced subgraph on S is connected and $\delta(S) \neq \emptyset$.

We note that condition (iii) is required in our analysis later, although the general problem of finding a local solution over S can be dealt with by solving the problem on each connected component of the induced subgraph on S individually. We remark that in this setup, we *do not* place any condition on b beyond having nonempty support. The entries in b may be positive or negative.

The global solution to the system $\mathcal{L}x = b$ satisfying the boundary condition b is a vector $x \in \mathbb{R}^V$ with

$$x(v) = \begin{cases} \sum_{u \sim v} \frac{x(u)}{\sqrt{d_v d_u}} & \text{if } v \in S \\ b(v) & \text{if } v \notin S \end{cases} \tag{3.1}$$

for a b-boundable subset S. The problem of interest is computing the *local* solution for the *restriction* of x to the subset S, denoted x_S.

The eigenvalues of $\mathcal{L}_S$ are called Dirichlet eigenvalues, denoted $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_s$, where $s = |S|$. It is easy to check (see [4]) that $0 < \lambda_i \leq 2$, since we assume $\delta(S) \neq \emptyset$. Thus, $\mathcal{L}_S^{-1}$ exists and is well defined. In fact, $s^{-3} < \lambda_1 \leq 1$.

Let $A_{S,\delta S}$ be the $s \times |\delta(S)|$ matrix by restricting the columns of A to $\delta(S)$ and rows to S. Requiring S to be a b-boundable subset ensures that the inverse $\mathcal{L}_S^{-1}$ exists [4]. Then the local solution is described exactly in the following theorem.

Theorem 3.2. *In a graph G, suppose b is a nontrivial vector in $\mathbb{R}^V$ and S is a b-boundable subset. Then, the local solution to the linear system $\mathcal{L}x = b$ satisfying the boundary condition b satisfies*

$$x_S = \mathcal{L}_S^{-1}(D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}). \tag{3.2}$$

Proof. The vector $b_1 := D_S^{-1/2} A_{\delta S} D_{\delta S}^{-1/2} b_{\delta S}$ is defined over the vertices of S, and give over the vertices of S by

$$b_1(v) = \sum_{u \in \delta(S), u \sim v} \frac{b(u)}{\sqrt{d_v d_u}}. \tag{3.3}$$

Also, the vector $\mathcal{L}_S x_S$ is given by, for $v \in S$,

$$\mathcal{L}_S x_S(v) = x(v) - \sum_{u \in S, u \sim v} \frac{x(u)}{\sqrt{d_v d_u}}. \tag{3.4}$$

By (3.1) and (3.2), we have

$$x_S(v) = \sum_{u \in S, u \sim v} \frac{x(u)}{\sqrt{d_v d_u}} + \sum_{u \in \delta(S), u \sim v} \frac{b(u)}{\sqrt{d_v d_u}},$$

and combining (3.3) and (3.4), we have that $x_S = \mathcal{L}_S^{-1} b_1$. □

3.1. Solving the local system with Green's function

For the remainder of this article we are concerned with the local solution x_S. We focus our discussion on the restricted space using the assumptions that the induced subgraph on S is connected and that $\delta(S) \neq \emptyset$. In particular, we consider the *Dirichlet heat kernel*, which is the heat kernel pagerank restricted to S.

The Dirichlet heat kernel is written by $\mathcal{H}_{S,t}$ and is defined as $\mathcal{H}_{S,t} = e^{-t\mathcal{L}_S}$. It is the symmetric version of $H_{S,t}$, where $H_{S,t} = e^{-t\Delta_S} = D_S^{-1/2} \mathcal{H}_{S,t} D_S^{1/2}$.

The spectral decomposition of $\mathcal{L}_S$ is

$$\mathcal{L}_S = \sum_{i=1}^{s} \lambda_i \mathbb{P}_i,$$

where $\mathbb{P}_i$ are the projections to the ith orthonormal eigenvectors. The Dirichlet heat kernel can be expressed as

$$\mathcal{H}_{S,t} = \sum_{i=1}^{s} e^{-t\lambda_i} \mathbb{P}_i.$$

Let $\mathcal{G}$ denote the inverse of $\mathcal{L}_S$. Namely, $\mathcal{G}\mathcal{L}_S = \mathcal{L}_S \mathcal{G} = I_S$. Then

$$\mathcal{G} = \sum_{i=1}^{s} \frac{1}{\lambda_i} \mathbb{P}_i. \tag{3.5}$$

From (3.5), we see that

$$\frac{1}{2} \leq \|\mathcal{G}\| \leq \frac{1}{\lambda_1}, \tag{3.6}$$

where $\|\cdot\|$ denotes the spectral norm. We call $\mathcal{G}$ the *Green's function*, and $\mathcal{G}$ can be related to $\mathcal{H}_{S,t}$ as follows:

Lemma 3.3. *Let $\mathcal{G}$ be the Green's function of a connected induced subgraph on $S \subset V$ with $s = |S|$. Let $\mathcal{H}_{S,t}$ be the Dirichlet heat kernel with respect to S. Then*

$$\mathcal{G} = \int_0^\infty \mathcal{H}_{S,t} \, \mathrm{d}t.$$

Proof. By our definition of the heat kernel,

$$\begin{aligned}\int_0^\infty \mathcal{H}_{S,t}\,\mathrm{d}t &= \int_0^\infty \left(\sum_{i=1}^{s} e^{-t\lambda_i}\mathbb{P}_i\right)\mathrm{d}t \\ &= \sum_{i=1}^{s}\left(\int_0^\infty e^{-t\lambda_i}\,\mathrm{d}t\right)\mathbb{P}_i \\ &= \sum_{i=1}^{s}\frac{1}{\lambda_i}\mathbb{P}_i \\ &= \mathcal{G}.\end{aligned}$$

□

Equipped with the Green's function, the solution (3.2) can be expressed in terms of the Dirichlet heat kernel. As a corollary to Theorem 3.2 we have the following.

Corollary 3.4. *In a graph G, suppose b is a nontrivial vector in $\mathbb{R}^V$ and S is a b-boundable subset. Then the local solution to the linear system $\mathcal{L}x = b$, satisfying the boundary condition b can be written as*

$$x_S = \int_0^\infty \mathcal{H}_{S,t} b_1\,\mathrm{d}t, \tag{3.7}$$

where $b_1 = D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$.

The computation of b_1 takes time proportional to the size of the edge boundary.

4. A LOCAL LINEAR SOLVER ALGORITHM WITH HEAT KERNEL PAGERANK

In the previous section, we saw how the local solution x_S to the system satisfying the boundary condition b can be expressed in terms of integrals of Dirichlet heat kernel in (3.7). In this section, we will show how these integrals can be well-approximated by sampling a finite number of values of Dirichlet heat kernel (Theorem 4.1) and Dirichlet heat kernel pagerank (Corollary 4.5). All norms $\|\cdot\|$ in this section are the L_2 norm.

Theorem 4.1. *Let G be a graph and $\mathcal{L}$ denote the normalized Laplacian of G. Let b be a nontrivial vector $b \in \mathbb{R}^V$ and S a b-boundable subset, and let $b_1 = D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$. Then the local solution x_S to the linear system $\mathcal{L}x = b$ satisfying the boundary condition b can be computed by sampling $\mathcal{H}_{S,t} b_1$ for $r = \gamma^{-2}\log(s\gamma^{-1})$ values. If $\hat{x}_S$ is the output of this process, the result has error bounded by*

$$\|x_S - \hat{x}_S\| = O\big(\gamma(\|b_1\| + \|x_S\|)\big)$$

with probability at least $1-\gamma$.

We prove Theorem 4.1 in two steps. First, we show how the integral (3.7) can be expressed as a finite Riemann sum without incurring much loss of accuracy in Lemma 4.2. Second, we show in Lemma 4.3 how this finite sum can be well-approximated by its expected value using a concentration inequality.

Lemma 4.2. *Let x_S be the local solution to the linear system $\mathcal{L}x = b$ satisfying the boundary condition b given in (3.7). Then, for $T = s^3 \log(s^3\gamma^{-1})$ and $N = T/\gamma$, the error incurred by taking a right Riemann sum is*

$$\left\| x_S - \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} \frac{T}{N} b_1 \right\| \leq \gamma(\|b_1\| + \|x_S\|),$$

where $b_1 = D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$.

Proof. First, we see that:

$$\begin{aligned} \|\mathcal{H}_{S,t}\| &= \left\| \sum_i e^{-t\lambda_i} \mathbb{P}_i \right\| \\ &\leq e^{-t\lambda_1} \left\| \sum_i \mathbb{P}_i \right\| \\ &= e^{-t\lambda_1} \end{aligned} \tag{4.1}$$

where λ_i are Dirichlet eigenvalues for the induced subgraph S. So the error incurred by taking a definite integral up to $t = T$ to approximate the inverse is the difference

$$\begin{aligned} \left\| x_S - \int_0^T \mathcal{H}_{S,t} b_1 \, \mathrm{d}t \right\| &= \left\| \int_T^\infty \mathcal{H}_{S,t} b_1 \, \mathrm{d}t \right\| \\ &\leq \int_T^\infty e^{-t\lambda_1} \|b_1\| \, \mathrm{d}t \\ &\leq \frac{1}{\lambda_1} e^{-T\lambda_1} \|b_1\| . \end{aligned}$$

Then by the assumption on T, the error is bounded by $\left\| x_S - \int_0^T \mathcal{H}_{S,t} b_1 \, \mathrm{d}t \right\| \leq \gamma \|b_1\|$.

Next, we approximate the definite integral in $[0, T]$ by discretizing it. That is, for a given γ, we choose $N = T/\gamma$ and divide the interval $[0, T]$ into N intervals of size T/N. Then a finite Riemann sum is close to the definite integral:

$$\begin{aligned} \left\| \int_0^T \mathcal{H}_{S,t} b_1 \, \mathrm{d}t - \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{T}{N} \right\| &\leq \gamma \left\| \int_0^T \mathcal{H}_{S,t} b_1 \, \mathrm{d}t \right\| \\ &\leq \gamma \|x_S\| . \end{aligned}$$

This gives a total error bounded by $\gamma(\|b_1\| + \|x_S\|)$. □

Lemma 4.3. *The sum $\sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{T}{N}$ can be approximated by sampling $r = \gamma^{-2} \log(s\gamma^{-1})$ values of $\mathcal{H}_{S,jT/N} b_1$ where j is drawn from $[1, N]$. With probability at least $1 - \gamma$, the result has multiplicative error at most γ.*

A main tool in our proof of Lemma 4.3 is the following matrix concentration inequality (see [5], also variations in [1], [9], [20], [12], [25]).

Theorem 4.4. *Let $X_1, X_2, \ldots, X_m$ be independent random $n \times n$ Hermitian matrices. Moreover, assume that $\|X_i - \mathbb{E}(X_i)\| \leq M$ for all i, and put $v^2 = \|\sum_i var(X_i)\|$. Let $X = \sum_i X_i$. Then for any $a > 0$,*

$$\Pr(\|X - \mathbb{E}(X)\| > a) \leq 2n \exp\left(-\frac{a^2}{2v^2 + 2Ma/3}\right),$$

where $\|\cdot\|$ denotes the spectral norm.

Proof of Lemma 4.3. Suppose without loss of generality that $\|b_1\| = 1$. Let Y be a random variable that takes on the vector $\mathcal{H}_{S,jT/N} b_1$ for every $j \in [1, N]$ with probability $1/N$. Then $\mathbb{E}(Y) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1$. Let $X = \sum_{i=1}^{r} X_j$ where each X_j is a copy of Y, so that $\mathbb{E}(X) = r\mathbb{E}(Y)$.

Now consider $\mathbb{Y}$ to be the random variable that takes on the projection matrix $\mathcal{H}_{S,jT/N} b_1 (\mathcal{H}_{S,jT/N} b_1)^T$ for every $j \in [1, N]$ with probability $1/N$, and $\mathbb{X}$ is the sum of r copies of $\mathbb{Y}$. Then we evaluate the expected value and variance of $\mathbb{X}$ as follows:

$$\begin{aligned}
\|\mathbb{E}(\mathbb{X})\| &= r\,\|\mathbb{E}(\mathbb{Y})\| \\
\|\mathrm{Var}(\mathbb{X})\| &= r\,\|\mathrm{Var}(\mathbb{Y})\| \leq \left\| \frac{r}{N} \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 (\mathcal{H}_{S,jT/N} b_1)^T \left\|\mathcal{H}_{S,jT/N} b_1\right\|^2 \right\| \\
&\leq r\,\|\mathbb{E}(\mathbb{Y})\|\,.
\end{aligned}$$

We now apply Theorem 4.4 to $\mathbb{X}$. We have

$$\begin{aligned}
\Pr\big(\|\mathbb{X} - \mathbb{E}(\mathbb{X})\| \geq \gamma\,\|\mathbb{E}(\mathbb{X})\|\big) &\leq 2s \exp\left(-\frac{\gamma^2\,\|\mathbb{E}(\mathbb{X})\|^2}{2\mathrm{Var}(\mathbb{X}) + \frac{2\gamma\|\mathbb{E}(\mathbb{X})\|M}{3}}\right) \\
&\leq 2s \exp\left(-\frac{\gamma^2 r^2\,\|\mathbb{E}(\mathbb{Y})\|}{r + 2\gamma r M/3}\right) \\
&\leq 2s \exp\left(-\frac{\gamma^2 r}{2}\right).
\end{aligned}$$

Therefore, we have $\Pr(\|\mathbb{X} - \mathbb{E}(\mathbb{X})\| \geq \gamma\,\|\mathbb{E}(\mathbb{X})\|) \leq \gamma$ if we choose $r \geq \gamma^{-2}\log(s\gamma^{-1})$. Further, this implies the looser bound:

$$\Pr\big(\|X - \mathbb{E}(X)\| \geq \gamma\,\|\mathbb{E}(X)\|\big) \leq \gamma.$$

Then, $\mathbb{E}(Y) = \frac{1}{r}\mathbb{E}(X)$ is close to $\frac{1}{r}X$ and

$$\left\| \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{1}{N} - \frac{1}{r} X \right\| \leq \gamma \left\| \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{1}{N} \right\|$$

$$\left\| \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{T}{N} - \frac{T}{r} X \right\| \leq \gamma \left\| \sum_{j=1}^{N} \mathcal{H}_{S,jT/N} b_1 \frac{T}{N} \right\|$$

with probability at least $1 - \gamma$, as claimed.

Proof of Theorem 4.1. Let X be the sum of r samples of $\mathcal{H}_{S,jT/N}b_1$ with j drawn from $[0, N]$, and let $\hat{x}_S = \frac{T}{r}X$. Then, combining Lemmas 4.2 and 4.3, we have

$$\begin{aligned}\|x_S - \hat{x}_S\| &\le \gamma\left(\|b_1\| + \|x_S\| + \left\|\sum_{j=1}^{N}\mathcal{H}_{S,jT/N}b_1\frac{T}{N}\right\|\right)\\ &\le O\big(\gamma(\|b_1\| + \|x_S\|)\big).\end{aligned}$$

By Lemma 4.3, this bound holds with probability at least $1-\gamma$.

The above analysis allows us to approximate the solution x_S by sampling $\mathcal{H}_{S,t}b_1$ for various t. The following corollary is similar to Theorem 4.1, except that we use the asymmetric version of the Dirichlet heat kernel, which we will need later for using random walks. In particular, we use Dirichlet heat kernel pagerank vectors. Dirichlet heat kernel pagerank is also defined in terms of a subset S whose induced subgraph is connected, and a vector $f \in \mathbb{R}^S$ by the following:

$$\rho_{S,t,f} = f^T H_{S,t}. \tag{4.2}$$

Corollary 4.5. *Let G be a graph and $\mathcal{L}$ denote the normalized Laplacian of G. Let b be a nontrivial vector $b \in \mathbb{R}^V$ and S be a b-boundable subset. Let $b_2 = (D_S^{-1/2}A_{S,\delta S}D_{\delta S}^{-1/2}b_{\delta S})^T D_S^{1/2}$. Then, the local solution x_S to the linear system $\mathcal{L}x = b$ satisfying the boundary condition b can be computed by sampling ρ_{S,t,b_2} for $r = \gamma^{-2}\log(s\gamma^{-1})$ values. If $\hat{x}_S$ is the output of this process, the result is error bounded by*

$$\|x_S - \hat{x}_S\| = O\big(\gamma(\|b_1\| + \|x_S\|)\big),$$

where $b_1 = D_S^{-1/2}A_{\delta S}D_{\delta S}^{-1/2}b_{\delta S}$, with probability at least $1-\gamma$.

Proof. First, we show how x_S can be given in terms of Dirichlet heat kernel pagerank.

$$\begin{aligned}x_S^T &= \int_0^\infty b_1^T\mathcal{H}_{S,t}\,\mathrm{d}t\\ &= \int_0^\infty b_1^T(D_S^{1/2}H_{S,t}D_S^{-1/2})\,\mathrm{d}t\\ &= \int_0^\infty b_2H_{S,t}D_S^{-1/2}\,\mathrm{d}t, \quad \text{where } b_2 = b_1^T D_S^{1/2}\\ &= \int_0^\infty \rho_{S,t,b_2}\,\mathrm{d}t\ D_S^{-1/2},\end{aligned}$$

and we have an expression similar to (3.7). Then, by Lemma 4.2, x_S^T is close to $\sum_{j=1}^N \rho_{S,jT/N,b_2}\frac{T}{N}D_S^{-1/2}$ with error bounded by $O(\gamma(\|b_1\| + \|x_S\|))$. From Lemma 4.3, this can be approximated to within $O(\gamma\,\|x_S\|)$ multiplicative error using $r = \gamma^{-2}\log(s\gamma^{-1})$ samples with probability at least $1-\gamma$. This gives total additive and multiplicative error within $O(\gamma)$. □

4.1. The local linear solver algorithm

We present an algorithm for computing a local solution to a Laplacian linear system with a boundary condition.

Theorem 4.6. *Let G be a graph and $\mathcal{L}$ denote the normalized Laplacian of G. Let b be a nontrivial vector $b \in \mathbb{R}^V$, S a b-boundable subset, and let $b_1 = D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$. For the linear system $\mathcal{L}x = b$, the solution x is required to satisfy the boundary condition b, and let x_S be the local solution. Then the approximate solution* $\mathtt{x}$ *output by the* Local Linear Solver *algorithm has an error bounded by*

$$\|x_S - \mathtt{x}\| = O\big(\gamma(\|b_1\| + \|x_S\|)\big)$$

with probability at least $1 - \gamma$.

Algorithm 1 *Local Linear Solver*

input: graph G, boundary vector $b \in \mathbb{R}^V$, subset $S \subset V$, solver error parameter $0 < \gamma < 1$.
output: an approximate local solution $\mathtt{x}$ with additive and multiplicative error γ to the local system $x_S = \mathcal{G}b_1$ satisyfing the boundary condition b.

$s \leftarrow |S|$
initialize a 0-vector $\mathtt{x}$ of dimension s
$b_1 \leftarrow D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$
$b_2 \leftarrow b_1^T D_S^{1/2}$
$T \leftarrow s^3 \log(s^3 \gamma^{-1})$
$N \leftarrow T/\gamma$
$r \leftarrow \gamma^{-2} \log(s\gamma^{-1})$
for $i = 1$ to r **do**
    draw j from $[1, N]$ uniformly at random
    $x_i \leftarrow \rho_{S, jT/N, b_2}$
    $\mathtt{x} \leftarrow \mathtt{x} + x_i$
end for
return $T/r \cdot \mathtt{x} D_S^{-1/2}$

Proof. The correctness of the algorithm follows from Corollary 4.5. □

The algorithm involves $r = \gamma^{-2} \log(s\gamma^{-1})$ Dirichlet heat kernel pagerank computations, so the running time is proportional to the time for computing $b_2 e^{-T\Delta_S}$ for $T = s^3 \log(s^3\gamma^{-1})$.

In the next sections, we discuss an efficient way to approximate a Dirichlet heat kernel pagerank vector and the resulting algorithm `GreensSolver` that returns approximate local solutions in sublinear time.

5. DIRICHLET HEAT KERNEL PAGERANK APPROXIMATION ALGORITHM

The definition of Dirichlet heat kernel pagerank in (4.2) is given in terms of a subset S and a vector $f \in \mathbb{R}^S$. Our goal is to express this vector as the stationary distribution of random walks on the graph in order to design an efficient approximation algorithm.

Dirichlet heat kernel pagerank is defined over the vertices of a subset S as follows:

$$\begin{aligned}\rho_{S,t,f} &= f^T H_{S,t} = f^T e^{-t\Delta_S} = f^T e^{-t(I_S - P_S)} \\ &= \sum_{k=0}^{\infty} e^{-t} \frac{t^k}{k!} f^T P_S^k.\end{aligned}$$

That is, it is defined in terms of the transition probability matrix P_S – the restriction of P where P describes a random walk on the graph. We can interpret the matrix P_S as the transition probability matrix of the following so-called *Dirichlet random walk*: Move from a vertex u in S to a neighbor v with probability $1/d_u$. If v is not in S, abort the walk and ignore any probability movement. Because we consider the diffusion of probability only within the subset, any random walks that leave S cannot be allowed to return any probability to S. To prevent this, random walks that do not remain in S are ignored.

We recall some facts about random walks. First, if g is a probabilistic function over the vertices of G, then $g^T P^k$ is the probability distribution over the vertices after performing k random walk steps according to P starting from vertices drawn from g. Similarly, when f is a probabilistic function over S, $f^T P_S^k$ is the distribution after k Dirichlet random walk steps. Consider a Dirichlet random walk process in which the number of steps taken, k (where steps are taken according to a Dirichlet random walk as described above), is a Poisson random variable with mean t. That is, k steps are taken with probability $p_t(k) = e^{-t}\frac{t^k}{k!}$. Then, the Dirichlet heat kernel pagerank is the expected distribution of this process.

In order to use random walks for approximating Dirichlet heat kernel pagerank, we perform some preprocessing for general vectors $f \in \mathbb{R}^S$. Namely, we do separate computations for the positive and negative parts of the vector, and normalize each part to be a probability distribution.

Given a graph and a vector $f \in \mathbb{R}^S$, the algorithm `ApproxDirHKPR` computes vectors that ϵ-approximate the Dirichlet heat kernel pagerank $\rho_{S,t,f}$ satisfying the following criteria:

Definition 5.1. Let G be a graph and let $S \subset V$ be a subset of vertices. Let $f \in \mathbb{R}^S$ be a probability distribution vector over the vertices of S, and let $\rho_{S,t,f}$ be the Dirichlet heat kernel pagerank vector according to S, t, and f. Then, we say that $\nu \in \mathbb{R}^S$ is an *ϵ-approximate Dirichlet heat kernel pagerank vector* if

1. for every vertex v in the support of ν, $|\rho_{S,t,f}(v) - \nu(v)| \leq \epsilon \cdot \rho_{S,t,f}(v)$, and
2. for every vertex with $\nu(v) = 0$, it must be that $\rho_{S,t,f}(v) \leq \epsilon$.

When f is a general vector, an ϵ-approximate Dirichlet heat kernel pagerank vector has an additional additive error of $\epsilon\|f\|_1$ by scaling, where $\|\cdot\|_1$ denotes the L_1 norm.

For example, the zero-vector is an ϵ-approximate of any vector with all entries of value $< \epsilon$. We remark that for a vector f with L_1 norm 1, the Dirichlet heat kernel pagerank vector $\rho_{S,t,f}$ has at most $1/\epsilon$ entries with values at least ϵ. Thus, a vector that ϵ-approximates $\rho_{S,t,f}$ has support of size at most $1/\epsilon$.

The time complexity of `ApproxDirHKPR` is given in terms of random walk steps. As such, the analysis assumes access to constant-time queries returning (i) the destination of a random walk step, and (ii) a sample from a distribution.

Algorithm 2 `ApproxDirHKPR`(G, t, f, S, ϵ)

input: a graph G, $t \in \mathbb{R}^+$, vector $f \in \mathbb{R}^S$, subset $S \subset V$, error parameter $0 < \epsilon < 1$.
output: ρ, an ϵ-approximation of $\rho_{S,t,f}$.

```
s ← |S|
initialize 0-vector ρ of dimension s
f_+ ← the positive portion of f
f_− ← the negative portion of f so that f = f_+ − f_−
f'_+ ← f_+/‖f_+‖_1                ▶ normalize f_+ to be a probability distribution vector
f'_− ← f_−/‖f_−‖_1                ▶ normalize f_− to be a probability distribution vector
r ← (16/ε^3) log n
for r iterations do
    choose a starting vertex u_1 according to the distribution vector f'_+
    k ∼ Poiss(t)                  ▶ choose k with probability e^{−t} t^k/k!
    k ← min{k, t/ε}
    simulate k steps of a P = D^{−1}A random walk
    if the random walk leaves S then:
        do nothing for the rest of this iteration
    else
        let v_1 be the last vertex visited in the walk
        ρ[v_1] ← ρ[v_1] + ‖f_+‖_1
    end if
    choose a starting vertex u_2 according to the distribution vector f'_−
    k ∼ Poiss(t)                  ▶ choose k with probability e^{−t} t^k/k!
    k ← min{k, t/ε}
    simulate k steps of a P = D^{−1}A random walk
    if the random walk leaves S then:
        do nothing for the rest of this iteration
    else
        let v_2 be the last vertex visited in the walk
        ρ[v_2] ← ρ[v_2] + ‖g_−‖_1
    end if
end for

ρ ← ρ/r
return ρ
```

Theorem 5.2. *Let G be a graph and S a proper vertex subset such that the induced subgraph on S is connected. Let f be a vector $f \in \mathbb{R}^S$, $t \in \mathbb{R}^+$, and $0 < \epsilon < 1$. Then the algorithm* `ApproxDirHKPR`(G, t, f, S, ϵ) *outputs an ϵ-approximate Dirichlet heat kernel pagerank vector $\hat{\rho}_{S,t,f}$ with probability at least $1 - \epsilon$. The running time of* `ApproxDirHKPR` *is $O(\epsilon^{-4} t \log n)$, where the constant hidden in the big-O notation reflects the time to perform a random walk step.*

Our analysis relies on the usual Chernoff bounds restated below. They will be applied in a similar fashion as in [3].

Lemma 5.3. ([3]) *Let X_i be independent Bernoulli random variables with $X = \sum_{i=1}^{r} X_i$. Then,*

1. *for* $0 < \epsilon < 1$, $\Pr(X < (1-\epsilon)r\mathbb{E}(X)) < \exp(-\frac{\epsilon^2}{2} r\mathbb{E}(X))$
2. *for* $0 < \epsilon < 1$, $\Pr(X > (1+\epsilon)r\mathbb{E}(X)) < \exp(-\frac{\epsilon^2}{4} r\mathbb{E}(X))$
3. *for* $c \geq 1$, $\Pr(X > (1+c)r\mathbb{E}(X)) < \exp(-\frac{c}{2} r\mathbb{E}(X))$.

Proof of Theorem 5.2. For the sake of simplicity, we provide analysis for the positive part of the vector, $f := f_+$, noting that it is easily applied similarly to the negative part as well.

The vector $f' = f/\|f\|_1$ is a probability distribution, and the heat kernel pagerank $\rho'_{S,t,f} = \rho_{S,t,f}/\|f\|_1$ can be interpreted as a series of Dirichlet random walks in which, with probability $e^{-t}\frac{t^k}{k!}$, $f'^T P_S^k$, is contributed to $\rho'_{S,t,f}$. This is demonstrated by examining the coefficients of the terms, since

$$e^{-t}\sum_{k=0}^{\infty}\frac{t^k}{k!} = 1.$$

The probability of taking $k \sim Pois(t)$ steps such that $k \geq t/\epsilon$ is less than ϵ by Markov's inequality. Therefore, enforcing an upper bound of $K = t/\epsilon$ for the number of random walk steps taken is enough mixing time with probability at least $1-\epsilon$.

For $k \leq t/\epsilon$, our algorithm approximates $f'^T P_S^k$ by simulating k random walk steps according to P as long as the random walk remains in S. If the random walk ever leaves S, it is ignored. To be specific, let X_k^v be the indicator random variable defined by $X_k^v = 1$ if a random walk beginning from a vertex u drawn from $f' = f/\|f\|_1$ ends at vertex v in k steps without leaving S. Let X^v be the random variable that considers the random walk process ending at vertex v in *at most* k steps without leaving S. That is, X^v assumes the vector X_k^v with probability $e^{-t}\frac{t^k}{k!}$. Namely, we consider the combined random walk

$$X^v = \sum_{k \leq t/\epsilon} e^{-t}\frac{t^k}{k!}X_k^v.$$

Now, let $\rho(k)_{S,t,f}$ be the contribution to the heat kernel pagerank vector $\rho'_{S,t,f}$ of walks of length at most k. The expectation of each X^v is $\rho(k)_{S,t,f}(v)$. Then, by Lemma 5,

$$\begin{aligned}\Pr(X^v < (1-\epsilon)\rho(k)_{S,t,f}(v)\cdot r) &< \exp(-\rho(k)_{S,t,f}(v)r\epsilon^2/2)\\ &= \exp(-(8/\epsilon)\rho(k)_{S,t,f}(v)\log n)\\ &< n^{-4}\end{aligned}$$

for every component with $\rho'_{S,t,f}(v) > \epsilon$, since then $\rho(k)_{S,t,f}(v) > \epsilon/2$. Similarly,

$$\begin{aligned}\Pr(X^v > (1+\epsilon)\rho(k)_{S,t,f}(v)\cdot r) &< \exp(-\rho(k)_{S,t,f}(v)r\epsilon^2/4)\\ &= \exp(-(4/\epsilon)\rho(k)_{S,t,f}(v)\log n)\\ &< n^{-2}.\end{aligned}$$

We conclude the analysis for the support of $\rho'_{S,t,f}$ by noting that $\hat{\rho}_{S,t,f} = \frac{1}{r}X^v$, and we achieve an ϵ-multiplicative error bound for every vertex v with $\rho'_{S,t,f}(v) > \epsilon$ with probability at least $1 - O(n^{-2})$.

However, if $\rho'_{S,t,f}(v) \leq \epsilon$, by the third part of Lemma 5, $\Pr(\hat{\rho}_{S,t,f}(v) > 2\epsilon) \leq n^{-8/\epsilon^2}$. We conclude that, with high probability, $\hat{\rho}_{S,t,f}(v) \leq 2\epsilon$.

Finally, when f is not a probability distribution, the above applies to $f' = f/\|f\|_1$. Let $\hat{\rho}'_{S,t,f}$ be the output of the algorithm using $f' = f/\|f\|_1$ and $\rho'_{S,t,f}$ be the corresponding Dirichlet heat kernel pagerank vector $\rho_{S,t,f'}$. The full error of the Dirichlet heat kernel pagerank returned is

$$\begin{aligned}\|\rho_{S,t,f} - \hat{\rho}_{S,t,f}\|_1 &\leq \|\|f\|_1\rho'_{S,t,f} - \|f\|_1\hat{\rho}'_{S,t,f}\|_1 \\ &\leq \|f\|_1\|\rho'_{S,t,f} - \hat{\rho}'_{S,t,f}\|_1 \\ &\leq \epsilon\|f\|_1\|\rho'_{S,t,f}\|_1 \\ &= \epsilon\|f\|_1.\end{aligned}$$

For the running time, we use the assumptions that performing a random walk step and drawing from a distribution with finite support require constant time. These are incorporated in the random walk simulation, which dominates the computation. Therefore, for each of the r rounds, at most K steps of the random walk are simulated, giving a total of $rK = O(\frac{16}{\epsilon^3}\log n \cdot t/\epsilon) = \tilde{O}(t)$ queries.

6. THE `GreensSolver` ALGORITHM

Here we present the main algorithm, `GreensSolver`, for computing a solution to a Laplacian linear system with a boundary condition. It is the Local Linear Solver algorithmic framework combined with the scheme for approximating Dirichlet heat kernel pagerank. The scheme is an optimized version of the algorithm `ApproxDirHKPR` with a slight modification. We call the optimized version `SolverApproxDirHKPR`.

Definition 6.1. Define `SolverApproxDirHKPR`(G, t, f, S, ϵ) to be the algorithm `ApproxDirHKPR`(G, t, f, S, ϵ) with the following modification to lines 11 and 21 after drawing $k \sim Poiss(t)$:

$$k \leftarrow min\{k, 2t\}.$$

Namely, this modification limits the length of random walk steps to at most $2t$.

Theorem 6.2. *Let G be a graph and S a subset of size s. Let $T = s^3\log(s^3\gamma^{-1})$, and let $N = T/\gamma$ for some $0 < \gamma < 1$. Suppose j is a random variable drawn from $[1, \lfloor N \rfloor]$ uniformly at random and let $t = jT/N$. Then if $\epsilon \geq \gamma$, the algorithm `SolverApproxDirHKPR` returns a vector that ϵ-approximates $\rho_{S,t,f}$ with probability at least $1 - \epsilon$. Using the same query assumptions as Theorem 5.2, the running time of `SolverApproxDirHKPR` is $O(\epsilon^{-3}t\log n)$.*

We will use the following Chernoff bound for Poisson random variables.

Lemma 6.3. ([19]) *Let X be a Poisson random variable with parameter t. Then, if $x > t$,*

$$\Pr(X \geq x) \leq e^{x-t-x\log(x/t)}.$$

Proof of Theorem 6.2. Let k be a Poisson random variable with parameter t. Similar to the proof of Theorem 5.2, we use Lemma 6 to reason that

$$\begin{aligned}\Pr(k \geq 2t) &\leq e^{2t-t-2t\log(2t/t)} \\ &= e^{t(1-2\log 2)} \\ &\leq \epsilon,\end{aligned}$$

as long as $t \geq \frac{\log(\epsilon^{-1})}{1-2\log 2}$.

Let E be the event that $t < \frac{\log(\epsilon^{-1})}{1-2\log 2}$. The probability of E is

$$\begin{aligned}\Pr\left(jT/N < \frac{\log(\epsilon^{-1})}{1-2\log 2}\right) &= \Pr\left(j < \frac{\log(\epsilon^{-1})}{\gamma(1-2\log 2)}\right) \\ &= \frac{\log(\epsilon^{-1})}{(1-2\log 2)s^3\log(s^3\gamma^{-1})},\end{aligned}$$

which is less than ϵ as long as $\epsilon \geq (\frac{\gamma}{s^3})^{(1-2\log 2)\epsilon s^3}$. This holds when $\epsilon \geq \gamma$.

As before, the algorithm consists of r rounds of random walk simulation, in which each walk is at most $2t$. The algorithm, therefore, makes $r \cdot 2t = \epsilon^{-3}32t\log n$ queries, requiring $O(\epsilon^{-3}t\log n)$ time.

Below we give the algorithm `GreensSolver`. The algorithm is identical to *Local Linear Solver* with the exception of line 10, where we use the approximation algorithm `SolverApproxDirHKPR` for Dirichlet heat kernel pagerank computation.

Algorithm 3 `GreensSolver`$(G, b, S, \gamma, \epsilon)$

input: graph G, boundary vector $b \in \mathbb{R}^V$, subset $S \subset V$, solver error parameter $0 < \gamma < 1$, Dirichlet heat kernel pagerank error parameter $0 < \epsilon < 1$.
output: an approximate local solution $\mathtt{x}$ to the local system $x_S = \mathcal{G}b_1$ satisyfing the boundary condition b.

1: $s \leftarrow |S|$
2: initialize a 0-vector $\mathtt{x}$ of dimension s
3: $b_1 \leftarrow D_S^{-1/2}A_{S,\delta S}D_{\delta S}^{-1/2}b_{\delta S}$
4: $b_2 \leftarrow b_1^T D_S^{1/2}$
5: $T \leftarrow s^3\log(s^3\gamma^{-1})$
6: $N \leftarrow T/\gamma$
7: $r \leftarrow \gamma^{-2}\log(s\gamma^{-1})$
8: **for** $i = 1$ to r **do**
9: draw j from $[1, N]$ uniformly at random
10: $x_i \leftarrow$ `SolverApproxDirHKPR`$(G, jT/N, b_2, S, \epsilon)$
11: $\mathtt{x} \leftarrow \mathtt{x} + x_i$
12: **end for**
13: **return** $T/r \cdot \mathtt{x}D_S^{-1/2}$

Theorem 6.4. *Let G be a graph and $\mathcal{L}$ denote the normalized Laplacian of G. Let b be a nontrivial vector $b \in \mathbb{R}^V$ and S be a b-boundable subset, and let $b_1 = D_S^{-1/2} A_{S,\delta S} D_{\delta S}^{-1/2} b_{\delta S}$. For the linear system $\mathcal{L}x = b$, the solution x is required to satisfy the boundary condition b, and let x_S be the local solution. Then, the approximate solution $\mathtt{x}$ output by the algorithm* `GreensSolver` *satisfies the following:*

(i) The error of $\mathtt{x}$ is $\|x_S - \mathtt{x}\| = O(\gamma(\|b_1\| + \|x_S\|) + \epsilon\|b_2\|_1)$ with probability at least $1 - \gamma$,

(ii) The running time of `GreensSolver` *is $O(\gamma^{-2}\epsilon^{-3}s^3 \log^2(s^3\gamma^{-1}) \log n)$, where the big-O constant reflects the time to perform a random walk step, plus additional preprocessing time $O(|\partial(S)|)$, where $\partial(S)$ denotes the edge boundary of S.*

Proof. The error of the algorithm using true Dirichlet heat kernel pagerank vectors is $O(\gamma(\|b_1\| + \|x_S\|))$ by Corollary 4.5, so to prove (i) we address the additional error of vectors output by the approximation of `SolverApproxDirHKPR`. By Theorem 6.2, `SolverApproxDirHKPR` outputs an ϵ-approximate Dirichlet heat kernel pagerank vector with probability at least $1 - \epsilon$. Let $\hat{\rho}_{S,t,f}$ be the output of an arbitrary run of `SolverApproxDirHKPR`(G, t, f, S, ϵ). Then $\left\|\rho_{S,t,f} - \hat{\rho}_{S,t,f}\right\| \leq \epsilon(\|\rho_{S,t,f'}\|_1 + \|f\|_1) = \epsilon\|f\|_1$ by the definition of ϵ-approximate Dirichlet heat kernel pagerank vectors, where $f' = f/\|f\|_1$ is the normalized vector f. This means that the total error of `GreensSolver` is

$$\|x_S - \mathtt{x}\| \leq O\left(\gamma(\|b_1\| + \|x_S\|)\right) + \epsilon\|b_2\|_1.$$

Next we prove (ii). The algorithm makes $r = \gamma^{-2}\log(s\gamma^{-1})$ sequential calls to `SolverApproxDirHKPR`. The maximum possible value of t is $T = s^3 \log(s^3\gamma^{-1})$, so any call to `SolverApproxDirHKPR` is bounded by $O(\epsilon^{-3}s^3 \log(s^3\gamma^{-1}) \log n)$. Thus, the total running time is $O(\gamma^2\epsilon^{-3}s^3 \log^2(s^3\gamma^{-1}) \log\, n)$.

The additional preprocessing time of $O(|\partial(S)|)$ is for computing the vectors b_1 and b_2; these may be computed as a preliminary procedure. □

We note that the running time is a sequential running time attained by calling `SolverApproxDirHKPR` r times. However, by calling these in r parallel processes, the algorithm has a parallel running time, which is simply the same as that for `SolverApproxDirHKPR`.

6.1. Restricted range for approximation

Since `SolverApproxDirHKPR` promises only approximate values for vertices whose true Dirichlet heat kernel pagerank vector values are greater than ϵ, the `GreensSolver` algorithm can be optimized even further by preempting when this is the case.

Figure 2 illustrates how vector values drop as t gets large. The network is the same example network given in Section 2.2 and is further examined in the next section. We let t range from 1 to $T = s^3 \log(s^3\gamma^{-1}) \approx 108739$ for $\gamma = 0.01$ and compute Dirichlet heat kernel pagerank vectors $\rho_{S,t,f}$. The figure plots L_1 norms of the vectors as a solid line, and the absolute value of the maximum entry in the vector as a dashed line. In this example, no vector entry is larger than 0.01 for t as small as 250.

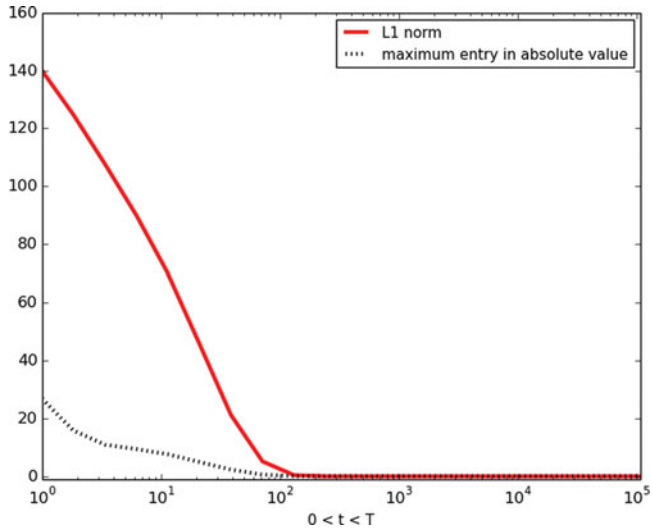

Figure 2 How support values of a Dirichlet heat kernel pagerank vector change for different values of $1 \leq t \leq T = s^3 \log(s^3\gamma^{-1})$. The solid line is the L_1 norm—the sum of all the support values—and the dashed line is the absolute value of the maximum entry in the vector. Note the x-axis is log-scale.

Suppose it is possible to know ahead of time whether a vector $\rho_{S,t,f}$ will have negligably small values for some value t. Then we could skip the computation of this vector and simply treat it as a vector of all zeros.

From (4.1), the norm of Dirichlet heat kernel pagerank vectors are monotone decreasing. Then, it is enough to choose a threshold value t' beyond which $\|\rho_{S,t',f}\|_1 < \epsilon$, since any ϵ-approximation will return all zeros, and treat this as a cutoff for actually executing the algorithm. An optimization heuristic is to compute `SolverApproxDirHKPR`(G, t, f, S, ϵ) only if t is less than this threshold value t'. Otherwise, we can add zeros (or do nothing). That is, replace line 10 in `GreensSolver` with the following:

if $jT/N < t'$ **then**
  $x_i \leftarrow$ `SolverApproxDirHKPR`$(G, jT/N, b_2, S, \epsilon)$
else
  do nothing
end if

From (4.1), a conservative choice for t' is $\frac{1}{\lambda_1}\log(\epsilon^{-1})$.

7. AN EXAMPLE ILLUSTRATING THE ALGORITHM

We return to our example to illustrate a run of the Green's solver algorithm for computing local linear solutions. The network is a small communication network of dolphins [18].

In this example, the subset has a good cluster, which makes it a good candidate for an algorithm in which computations are localized. Namely, it is ideal for `SolverApproxDirHKPR`, which promises good approximation for vertices that exceed a certain support threshold in terms of the error parameter ϵ. The support of the vector b

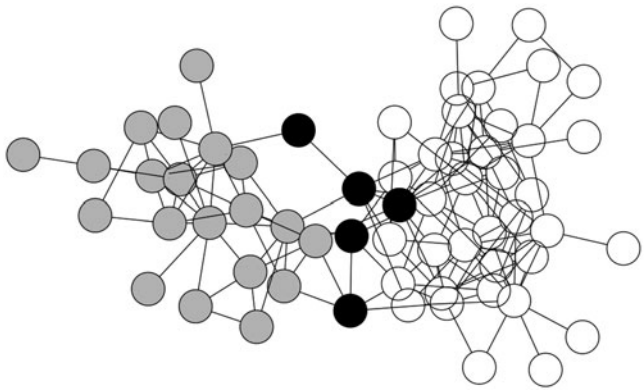

Figure 3 The values of the boundary vector plotted against the agent IDs given in Figure 1.

is limited to the set of leaders, which is the vertex boundary of the subset of followers, $l = \delta(f)$. The vector is plotted over the agents (vertices) in Figure 3.

Figure 4 plots the vector values of the heat kernel pagerank vector $\rho_{t,b_2'}$ over the full set of agents. Here, we use b_2', the n-dimensional vector:

$$b_2'(v) = \begin{cases} b_2(v) \text{ if } v \in S, \\ 0 \text{ otherwise,} \end{cases}$$

and $t = 50.0$. The components with largest absolute value are concentrated in the subset of followers over which we compute the local solution. This indicates that an output of `SolverApproxDirHKPR` will capture these values well.

7.1. Approximate solutions

In the following figures, we plot the results of calls to our approximation algorithms against the exact solution x_S using the boundary vector of Figure 3.

The solution x_S is computed by Theorem 3.2, and the appromimations are sample outputs of Local Linear Solver and `GreensSolver`, respectively. The exact values of x_S

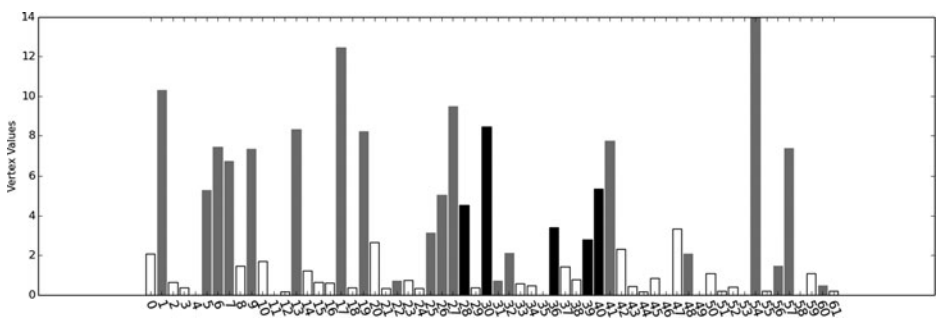

Figure 4 The node values of the full example communication network over a sample heat kernel pagerank vector. The grey bars correspond to the network of followers, the black to the leaders, and the white to the rest of the network.

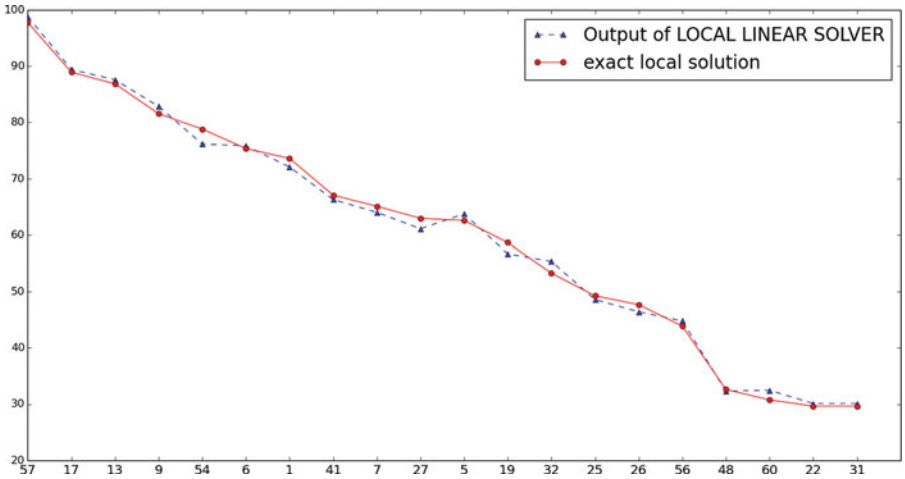

Figure 5 The results of a run of Local Linear Solver. Two vectors are plotted over IDs of agents in the subset. The circles are exact values of x_S, whereas the triangles are the approximate values returned by Local Linear Solver.

are represented by circles, and the approximate values by triangles in each case. Note that we permute the indices of the vertices in the solutions so that vector values in the exact solution, x_S are decreasing, for reading ease.[1]

The result of a sample call to Local Linear Solver with error parameter $\gamma = 0.01$ is plotted in Figure 5. The total relative error of this solution is $\frac{\|x_S - \hat{x}_S\|}{\|x_S\|} = 0.02$, and the absolute error $\|x_S - \hat{x}_S\|$ is within the error bounds given in Theorem 4.6. That is, $\|x_S - \hat{x}_S\| \leq \gamma\,(\|b_1\| + \|x_S\| + \|x_{rie}\|)$, where x_{rie} is the solution obtained by computing the full Riemann sum (as in Lemma 4.2).

The result of a sample call to `GreensSolver` with parameters $\gamma = 0.01, \epsilon = 0.1$ is plotted in Figure 6. In this case, the relative error is ≈ 2.05, but the absolute error meets the error bounds promised in Theorem 6.4 point (i). Specifically,

$$\|x_S - \hat{x}_S\| \leq (\gamma(\|b_1\| + \|x_S\| + \|x_{rie}\|) + \epsilon\|b_2\|_1)\,.$$

8. GENERAL REMARKS

Although we have focused our analysis on solving local linear systems with the normalized Laplacian $\mathcal{L}$ as the coefficient matrix, our methods can be extended to solve local linear systems expressed in terms of the Laplacian L as well. There are numerous applications involving solving such linear systems. Some examples are discussed in [6], and include computing effective resistance in electrical networks, computing maximum flow by interior point methods, describing the motion of coupled oscillators, and computing state in a network of communicating agents. In addition, we expect the method of approximating Dirichlet heat kernel pagerank in its own right to be useful in a variety of related applications.

[1]The results of these experiments as well as the source code are archived at `http://cseweb.ucsd.edu/~osimpson/localsolverexample.html`.

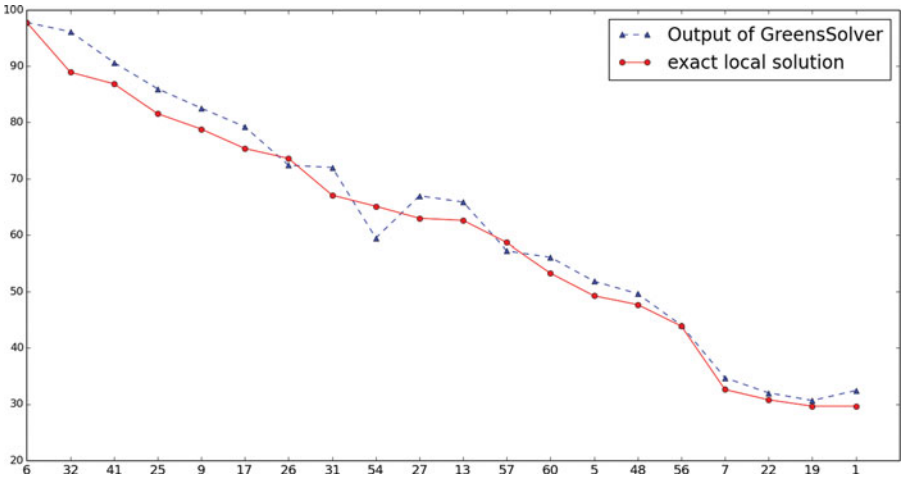

Figure 6 The results of a run of `GreensSolver` with $\gamma = 0.01, \epsilon = 0.1$.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their comments and suggestions. Their input has been immensely helpful in improving the presentation of the results and clarifying details of the algorithm.

REFERENCES

[1] R. Ahlswede and A. Winter. "Strong Converse for Identification via Quantum Channels." *IEEE Trans. Inform. Theory* 48:3 (2002), 569–579.

[2] G. E. Blelloch, A. Gupta, I. Koutis, G. L. Miller, and R. Peng. "Near Linear-Work Parallel SDD Solvers, Low-Diameter Decomposition, and Low-Stretch Subgraphs." In *Proceedings of the 23th ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 13–22. New York, NY: ACM, 2011.

[3] C. Borgs, M. Brautbar, J. T. Chayes, and S.H. Teng. "A Sublinear Time Algorithm for Pagerank Computations." In *WAW*, pp. 41–53. LNCS. Berlin, Heidelberg: Springer, 2012.

[4] F. Chung. *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1997.

[5] F. Chung and M. Radcliffe. *On the spectra of general random graphs*, The Electronic Journal of Combinatorics **18** (2011), P215.

[6] F. Chung and O. Simpson. "Solving Linear Systems with Boundary Conditions Using Heat Kernel Pagerank." *Algorithms and Models for the Web Graph*, pp. 203–219, LNCS 8305. Switzerland, Springer International, 2013.

[7] F. Chung and O. Simpson. "Computing Heat Kernel Pagerank and a Local Clustering Algorithm." To appear in *Proceedings of the 25th International Workshop on Combinatorial Algorithms*, LNCS. Berlin, Heidelberg: Springer, 2014.

[8] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. Rao, and S. C. Xu. "Solving SDD Linear Systems in Nearly $m \log^{1/2} n$ Time." In *STOC*. New York, NY: ACM, 2014.

[9] D. Cristofides and K. Markström. "Expansion Properties of Random Cayley Graphs and Vertex Transitive Graphs via Matrix Martingales." *Random Structures Algs*. 32:8 (2008), 88–100.

[10] G. E. Forsythe and R. A. Leibler. "Matrix Inversion by a Monte Carlo Method." *Mathematical Tables and Other Aids to Computation* 4:31 (1950), 127–129.

[11] K. D. Gremban, G. L. Miller, and M. Zagha. "Performance Evaluation of a New Parallel Preconditioner." In *Proceedings of the 9th International Parallel Processing Symposium*, pp. 65–69. IEEE, 1995.

[12] D. Gross. "Recovering Low-Rank Matrices from Few Coeffcients in Any Basis." *IEEE Trans. Inform. Theory* 57 (2011), 1548–1566.

[13] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. "A Simple, Combinatorial Algorithm for Solving SDD Systems in Nearly-Linear Time." In *STOC*, 911–920. New York, NY: ACM, 2013.

[14] I. Koutis and G. L. Miller. "A Linear Work $o(n^{1/6})$ Time Algorithm for Solving Planar Laplacians." In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1002–1011. ACM-SIAM, 2007.

[15] I. Koutis, G. L. Miller, and R. Peng. "Approaching Optimality for Solving SDD Linear Systems." In *FOCS (2010)*, pp. 235–244. IEEE Computer Society, 2010.

[16] I. Koutis, G. L. Miller, and R. Peng. "A Nearly-$m \log n$ Time Solver for SDD Linear Systems." In *FOCS (2011)*, 590–598. IEEE Computer Society, 2011.

[17] Y. T. Lee and A. Sidford. "Efficient Accelerated Coordinate Descent Methods and Faster Algorithms for Solving Linear Systems." In *FOCS (2013)*. IEEE Computer Society, 2013.

[18] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. "The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-Lasting Associations." *Behavioral Ecology and Sociobiology* 54 (2003), 396–405.

[19] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge, UK: Cambridge University Press, 2005.

[20] R. I. Oliveira. "Concentration of the Adjacency Matrix and of the Laplacian in Random Graphs with Independent Edges." arXiv preprint arXiv:0911.0600 (2009).

[21] R. Peng and D. A. Spielman. "An Efficient Parallel Solver for SDD Linear Systems." arXiv:1311.3286. (November 2013).

[22] S. Sachdeva and N. K. Vishnoi. "Matrix Inversion Is as Easy as Exponentiation." arXiv preprint arXiv:1305.0526 (2013).

[23] D. A. Spielman. "Algorithms, Graph Theory, and Linear Equations in Laplacian Matrices." In *Proceedings of the International Congress of Mathematicians* 4 (2010), 2698–2722.

[24] D. A. Spielman and S. H. Teng. "Nearly-Linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems." In *STOC (2004)*, pp. 81–90. New York, NY: ACM, 2004.

[25] J. A. Tropp. "User-Friendly Tail Bounds for Sums of Random Matrices." *Foundations of Computational Mathematics* 12:4 (2012), 389–434.

[26] P. M. Vaidya. "Solving Linear Equations with Symmetric Diagonally Dominant Matrices by Constructing Good Preconditioners." A talk based on this manuscript was presented as the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991.

[27] N. K. Vishnoi. *$Lx = b$ Laplacian solvers and their algorithmic applications*. Foundations and Trends in Theoretical Computer Science 8, nos. 1–2 (2012), 1–141.