

# Computing the Stationary Distribution, *Locally*

Christina E. Lee, Asuman Ozdaglar, Devavrat Shah

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139,  
celeee@mit.edu, asuman@mit.edu, devavrat@mit.edu

Computing the stationary distribution of a large finite or countably infinite state space Markov Chain has become central to many problems such as statistical inference and network analysis. Standard methods involve large matrix multiplications as in power iteration, or simulations of long random walks, as in Markov Chain Monte Carlo (MCMC). For both methods, the convergence rate is difficult to determine for general Markov chains. Power iteration is costly, as it is global and involves computation at every state. In this paper, we provide a novel local algorithm that answers whether a chosen state in a Markov chain has stationary probability larger than some  $\Delta \in (0, 1)$ , and outputs an estimate of the stationary probability for itself and other nearby states. Our algorithm runs in constant time with respect to the Markov chain, using information from a *local neighborhood* of the state on the graph induced by the Markov chain, which has constant size relative to the state space. The multiplicative error of the estimate is upper bounded by a function of the mixing properties of the Markov chain. Simulation results show Markov chains for which this method gives tight estimates.

*Key words:* Markov chains, stationary distribution, local algorithms, network centralities

---

## 1. Introduction

Computing the stationary distribution of a Markov chain with a very large state space (finite, or countably infinite) has become central to statistical inference. The ability to tractably simulate Markov chains along with its generic applicability has made Markov Chain Monte Carlo (MCMC) arguably one of the top algorithms of the twentieth century (Cipra 2000). However, MCMC and its variations suffer from limitations in large state spaces, as they involve sampling states from long random walks over the entire state space (Metropolis et al. 1953, Hastings 1970). It is difficult to determine for general Markov chains when the algorithm has walked “long enough” to produce reasonable approximations for the stationary distribution. This has motivated the development of super-computation capabilities – be it nuclear physics (Semkow et al. 2006, Chapter 8), Google’s computation of PageRank (Page et al. 1999), or stochastic simulation at-large (Assmussen and Glynn 2010).

Stationary distributions of Markov chains are also central to *network analysis*. Networks have become ubiquitous representations for capturing interactions and relationships between entities across many disciplines, including social interactions between individuals, interdependence between

financial institutions, hyper-link structure between web-pages, or correlations between distinct events. Many decision problems over networks rely on information about the importance of different nodes as quantified by network centrality measures. Network centrality measures are functions assigning “importance” values to each node in the network. The stationary distribution of specific random walks on these underlying networks are used as network centrality measures in many settings. A few examples include PageRank: which is commonly used in Internet search algorithms (Page et al. 1999), the Bonacich centrality and eigencentrality measures: encountered in the analysis of social networks (Newman 2010, Candogan et al. 2012, Chasparis and Shamma 2010), rumor centrality: utilized for finding influential individuals in social media like Twitter (Shah and Zaman 2011), and rank centrality: used to find a ranking over items within a network of pairwise comparisons (Negahban et al. 2012).

### 1.1. Contributions

In this paper, we provide a novel algorithm that addresses these limitations<sup>1</sup>. Our algorithm answers the following question: for a given node  $i$  of a countable state space Markov chain, is the stationary probability of  $i$  larger than a given threshold  $\Delta \in (0, 1)$ , and can we approximate it? For chosen parameters  $\Delta, \epsilon$ , and  $\alpha$ , our algorithm guarantees that for nodes such that the estimate  $\hat{\pi}_i < \Delta/(1 + \epsilon)$ , the true value  $\pi_i$  is also less than  $\Delta$  with probability at least  $1 - \alpha$ . In addition, if  $\hat{\pi}_i \geq \Delta/(1 + \epsilon)$ , with probability at least  $1 - \alpha$ , the estimate is within an  $\epsilon$  times  $Z_{\max}(i)$  multiplicative factor away from the true  $\pi_i$ , where  $Z_{\max}(i)$  is effectively a “local mixing time” for  $i$  derived from the fundamental matrix of the transition probability matrix  $P$ . The algorithm also gives estimates for other nodes  $j$  within the vicinity of node  $i$ . The estimation error depends on the mixing and connectivity properties of node  $i$  and node  $j$ .

The running time of the algorithm is upper bounded by  $\tilde{O}(\ln(1/\alpha)/\epsilon^3\Delta)$ , which is constant with respect to the Markov chain<sup>2</sup>. Our algorithm uses only a “local neighborhood” of the state  $i$ , defined with respect to the Markov graph, which is constructed using the transition probabilities between states. Stopping conditions are easy to verify and have provable performance guarantees. Its construction relies on a basic property: the stationary probability of each node is inversely proportional to the mean of its “return time.” Therefore, we sample return times to the node and use the empirical average as an estimate. Since return times can be arbitrarily long, we truncate sample return times at a chosen threshold. Hence, our algorithm is a *truncated Monte Carlo* method. We also use these samples to obtain estimates to other nodes  $j$  by observing the frequency of visits to node  $j$  within a return path to node  $i$ .

We utilize the exponential concentration of return times in Markov chains to establish theoretical guarantees for the algorithm. For countably infinite state space Markov chains, we build upon a

result by Hajek (1982) on the concentration of certain types of hitting times to derive concentration of return times to a given node. We use these concentration results to upper bound the estimation error and the algorithm runtime as a function of the truncation threshold and the mixing properties of the graph. For graphs that mix quickly, the distribution over return times concentrates more sharply around its mean, resulting in tighter performance guarantees. We illustrate the wide applicability of our local algorithm for computing network centralities and stationary distributions of queuing models.

## 1.2. Related Literature

We provide a brief overview of the standard methods used for computing stationary distributions.

**1.2.1. Monte Carlo Markov Chain** Monte Carlo Markov chain methods involve simulating long random walks over a carefully designed Markov chain in order to obtain samples from a target stationary distribution (Metropolis et al. 1953, Hastings 1970). After the length of this random walk exceeds the mixing time, the distribution over the current state of the random walk will be a close approximation to the stationary distribution. Thus, the observed current state of the random walk is used as an approximate sample from  $\pi$ . This process is repeated many times to collect independent samples from  $\pi_i$ . Articles by Diaconis and Saloff-Coste (1998) and Diaconis (2009) provide a summary of the major development from probability theory perspective.

The majority of work following the initial introduction of this method involves analyzing the convergence rates and mixing times of the random walk over different Markov chains (Aldous and Fill 1999, Levin et al. 2009). Techniques involve spectral analysis or coupling arguments. Graph properties such as conductance provide ways to characterize *the spectrum of the graph*. Most results are limited to reversible finite state space Markov chains, which are equivalent to random walks on weighted undirected graphs. For general non-reversible countable state space Markov chains, little is known about the mixing time, and thus many practical Markov chains lack precise convergence rate bounds.

**1.2.2. Power Iteration** The power-iteration method (see Golub and Van Loan (1996), Stewart (1994), Koury et al. (1984)) is an equally old and well-established method for computing leading eigenvectors of matrices. Given a matrix  $A$  and a seed vector  $x_0$ , recursively compute iterates  $x_{t+1} = Ax_t / \|Ax_t\|$ . If matrix  $A$  has a single eigenvalue that is strictly greater in magnitude than all other eigenvalues, and if  $x_0$  is not orthogonal to the eigenvector associated with the dominant eigenvalue, then a subsequence of  $x_t$  converges to the eigenvector associated with the dominant eigenvalue. Recursive multiplications involving large matrices can become expensive very fast as the matrix grows. When the matrix is sparse, computation can be saved by implementing it through

‘message-passing’ techniques; however it still requires computation to take place at every node in the state space. The convergence rate is governed by the spectral gap, or the difference between the two largest eigenvalues. Techniques used for analyzing the spectral gap and mixing times as discussed above are also used in analyzing the convergence of power iteration. For large Markov chains, the mixing properties may scale poorly with the size, making it difficult to obtain good estimates in a reasonable amount of time. As before, most results only pertain to reversible Markov chains.

In the setting of computing PageRank, there have been efforts to modify the algorithm to execute power iteration over local subsets of the graph and combine the results to obtain estimates for the global PageRank. These methods rely upon key assumptions on the underlying graph, which are difficult to verify. Kamvar et. al. observed that there may be obvious ways to partition the web graph (i.e. by domain names) such that power iteration can be used to estimate the local PageRank within these partitions (Kamvar et al. 2003). They use heuristics to estimate the relative weights of these partitions, and combine the local PageRank within each partition according to the weights to obtain an initial estimate for PageRank. This initial estimate is used to initialize the power iteration method over the global Markov chain, with the hope that this initialization may speed up convergence. Chen et. al. proposed a method for estimating the PageRank of a subset of nodes given only the local neighborhood of this subset (Chen et al. 2004). Their method uses heuristics such as weighted in-degree as estimates for the PageRank values of nodes on the boundary of the given neighborhood. After fixing the boundary estimates, standard power iteration is used to obtain estimates for nodes within the local neighborhood. The error in this method depends on how close the true PageRank of nodes on the boundary correspond to the heuristic guesses such as weighted in-degree. Unfortunately, we rarely have enough information to make accurate heuristic guesses of these boundary nodes.

**1.2.3. Computing PageRank Locally** There has been much recent effort to develop local algorithms for computing PageRank for the web graph. Given a directed graph of  $n$  nodes with an  $n \times n$  adjacency matrix  $A$  (i.e.,  $A_{ij} = 1$  if  $(i, j) \in E$  and 0 otherwise), the PageRank vector  $\pi$  is given by the stationary distribution of a Markov chain over  $n$  states, whose transition matrix  $P$  is given by

$$P = (1 - \beta)D^{-1}A + \beta\mathbf{1} \cdot r^T. \quad (1)$$

$D$  denotes the diagonal matrix whose diagonal entries are the out-degrees of the nodes;  $\beta \in (0, 1)$  is a fixed scalar; and  $r$  is a fixed probability vector over the  $n$  nodes<sup>3</sup>. In each step the random walk with probability  $(1 - \beta)$  chooses one of the neighbors of the current node equally likely, and with

probability  $\beta$  chooses any of the nodes in the graph according to  $r$ . Thus, the PageRank vector  $\pi$  satisfies

$$\pi^T = \pi^T P = (1 - \beta)\pi^T D^{-1}A + \beta r^T \quad (2)$$

where  $\pi^T \cdot \mathbf{1} = 1$ . This definition of PageRank is also known as personalized PageRank, because  $r$  can be tailored to the personal preferences of a particular web surfer. When  $r = \frac{1}{n} \cdot \mathbf{1}$ , then  $\pi$  equals the standard global PageRank vector. If  $r = e_i$ , then  $\pi$  describes the personalized PageRank that jumps back to node  $i$  with probability  $\beta$  in every step<sup>4</sup>.

Computationally, the design of local algorithms for computing the personalized PageRank has been of interest since its discovery. Most of the algorithms and analyses crucially rely on the specific structure of the random walk describing PageRank:  $P$  decomposes into a natural random walk matrix  $D^{-1}A$ , and a rank-1 matrix  $\mathbf{1} \cdot r^T$ , with strictly positive weights  $(1 - \beta)$  and  $\beta$  respectively, cf. (1). Jeh and Widom (2003) and Haveliwala (2003) observed a key linearity relation – the global PageRank vector is the average of the  $n$  personalized PageRank vectors corresponding to those obtained by setting  $r = e_i$  for  $1 \leq i \leq n$ . That is, these  $n$  personalized PageRank vectors centered at each node form a basis for all personalized PageRank vectors, including the global PageRank. Therefore, the problem boils down to computing the personalized PageRank for a given node. Fogaras et al. (2005) used the fact that for the personalized PageRank centered at a given node  $i$  (i.e.,  $r = e_i$ ), the associated random walk has probability  $\beta$  at every step to jump back to node  $i$ , “resetting” the random walk. The distribution over the last node visited before a “reset” is equivalent to the personalized PageRank vector corresponding to node  $i$ . Therefore, they propose an algorithm which samples from the personalized PageRank vector by simulating short geometric-length random walks beginning from node  $i$ , and recording the last visited node of each sample walk. The performance of the estimate can be established using standard concentration results.

Subsequent to the key observations mentioned above, Avrachenkov et al. (2007) surveyed variants to Fogaras’ random walk algorithm, such as computing the frequency of visits to nodes across the sample path rather than only the end node. Bahmani et al. (2010) addressed how to incrementally update the PageRank vector for dynamically evolving graphs, or graphs where the edges arrive in a streaming manner. Das Sarma *et al.* extended the algorithm to streaming graph models (Sarma et al. 2011), and distributed computing models (Sarma et al. 2012), “stitching” together short random walks to obtain longer samples, and thus reducing communication overhead. More recently, building on the same sets of observation, Borgs et al. (2012) provided a sublinear time algorithm for estimating global PageRank using multi-scale matrix sampling. They use geometric-length random walk samples, but do not require samples for all  $n$  personalized PageRank vectors. The algorithm

returns a set of “important” nodes such that the set contains all nodes with PageRank greater than a given threshold,  $\Delta$ , and does not contain any node with PageRank less than  $\Delta/c$  with probability  $1 - o(1)$ , for a given  $c > 1$ . The algorithm runs in time  $\tilde{O}(n/\Delta)$ .

Andersen et al. (2007) designed a backward variant of these algorithms. Previously, to compute the global PageRank of a specific node  $j$ , we would average over all personalized PageRank vectors. The algorithm proposed by Andersen et al. estimates the global PageRank of a node  $j$  by approximating the “contribution vector”, i.e. estimating for the  $j^{\text{th}}$  coordinates of the personalized PageRank vectors that contribute the most to  $\pi_j$ .

All of these algorithms rely on the crucial property that the random walk has renewal time that is distributed geometrically with constant parameter  $\beta > 0$  that does not scale with graph size  $n$ . This is because the transition matrix  $P$  decomposes according to (1), with a fixed  $\beta$ . In general, the transition matrix of any irreducible, positive-recurrent Markov chain will not have such a decomposition property (and hence known renewal time), making the above algorithms inapplicable in general.

## 2. Preliminaries: Markov Chains

Consider a discrete-time, irreducible, positive-recurrent Markov chain  $\{X_t\}_{t \geq 0}$  on a countable state space  $\Sigma$ . Let  $P$  denote the transition probability matrix, and let  $P_{xy}^{(n)}$  denote the value of entry  $(x, y)$  in the matrix  $P^n$ . If the state space is countably infinite, then  $P : \Sigma \times \Sigma \rightarrow [0, 1]$  is a function such that for all  $x, y \in \Sigma$ ,

$$P_{xy} = \mathbb{P}(X_{t+1} = y | X_t = x).$$

Similarly,  $P_{xy}^{(n)}$  is defined for all  $x, y \in \Sigma$  to be

$$P_{xy}^{(n)} \triangleq \mathbb{P}(X_n = y | X_0 = x).$$

The *stationary distribution* is a function  $\pi : \Sigma \rightarrow [0, 1]$  such that  $\sum_{i \in \Sigma} \pi_i = 1$  and  $\pi_i = \sum_{j \in \Sigma} \pi_j P_{ji}$  for all  $i \in \Sigma$ . The stationary distribution is the quantity that we are interested in estimating, with a focus on identifying states with large stationary probabilities.

The Markov chain can be visualized as a random walk over a weighted directed graph  $G = (\Sigma, E, P)$ , where  $\Sigma$  is the set of nodes,  $E = \{(i, j) \in \Sigma \times \Sigma : P_{ij} > 0\}$  is the set of edges, and  $P$  describes the weights of the edges. We refer to  $G$  as the *Markov chain graph*. The state space  $\Sigma$  is assumed to be either finite or countably infinite. If it is finite, let  $n = |\Sigma|$  denote the number of nodes in the graph<sup>5</sup>.

We assume throughout this paper that the Markov chain  $X_t$  is irreducible<sup>6</sup> and positive recurrent<sup>7</sup>. This guarantees that there exists a unique stationary distribution.

We define the *local neighborhood of size  $r$*  around node  $i \in \Sigma$  to be the subgraph  $G' = (\Sigma', E')$  induced by the set of nodes  $\Sigma' = \{j \in \Sigma : d_G(i, j) \leq r\}$ , where  $d_G(i, j)$  is the length of the shortest directed path (in terms of number of edges) from  $i$  to  $j$  in  $G$ . In this work, we consider the setting where the state space is large or even countably infinite. We will restrict our algorithm to accessing information within a local neighborhood around a fixed node  $i$ .

In order to obtain estimates for the stationary probability, our algorithm will generate sample sequences of the Markov chain by simulating a random walk on the graph. These sample sequences allow us to observe return times  $T_i$  and visit frequencies  $F_j$  to different nodes, where  $T_i$  and  $F_j$  are defined as:

$$T_i \triangleq \inf\{t \geq 1 : X_t = i\}, \quad (3)$$

and

$$F_j \triangleq \sum_{t=1}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} = \sum_{t=1}^{T_i} \mathbb{1}\{X_t = j\}.$$

Throughout this paper, we denote  $\mathbb{E}_i[\cdot] \triangleq \mathbb{E}[\cdot | X_0 = i]$ , and  $\mathbb{P}_i(\cdot) \triangleq \mathbb{P}(\cdot | X_0 = i)$ . The following property of Markov chains is crucial to our algorithm and analysis. It relates the stationary probabilities to the expected values for  $T_i$  and  $F_j$ .

LEMMA 1. (*Meyn and Tweedie 1993*) *An irreducible positive recurrent Markov chain has a unique stationary distribution  $\pi$  with the following form:*

(a) *For any fixed  $i \in \Sigma$ ,*

$$\pi_j = \frac{\mathbb{E}_i[F_j]}{\mathbb{E}_i[T_i]}, \quad j \in \Sigma.$$

(b) *An equivalent expression for this distribution is*

$$\pi_i = \frac{1}{\mathbb{E}_i[T_i]}.$$

We will utilize this property to show that given samples of  $T_i$  and  $F_j$ , we can obtain good estimates for the stationary probabilities. However, there are some Markov chains that are easier to compute over than others. In the finite state space setting, the performance guarantees for our algorithm will be given as a function of the maximal hitting time  $H_i$ , and the fundamental matrix  $Z$ . This measures how well connected the graph is globally. The *maximal hitting time* to a node  $i$  in a finite state space Markov chain is defined as

$$H_i \triangleq \max_{j \in \Sigma} \mathbb{E}_j[T_i]. \quad (4)$$

The *fundamental matrix*  $Z$  of a finite state space Markov chain is

$$Z \triangleq \sum_{t=0}^{\infty} (P^{(t)} - \mathbf{1}\pi^T) = (I - P + \mathbf{1}\pi^T)^{-1},$$

i.e., the entries of the fundamental matrix  $Z$  are defined by

$$Z_{jk} \triangleq \sum_{t=0}^{\infty} (P_{jk}^{(t)} - \pi_k).$$

Since  $P_{jk}^{(t)}$  denotes the probability that a random walk beginning at node  $j$  is at node  $k$  after  $t$  steps,  $Z_{jk}$  represents how quickly the probability mass at node  $k$  from a random walk beginning at node  $j$  converges to  $\pi_k$ . We will use the following property, stated by Aldous and Fill (1999), to relate entries in the fundamental matrix to expected return times.

LEMMA 2. For  $j \neq k$ ,

$$\mathbb{E}_j[T_k] = \frac{Z_{kk} - Z_{jk}}{\pi_k}.$$

We define  $Z_{\max}(i) \triangleq \max_{k \in \Sigma} |Z_{ki}|$ . The relationship between  $Z_{\max}(i)$  and  $H_i$  is described by

$$Z_{\max}(i) \leq \pi_i H_i \leq 2Z_{\max}(i).$$

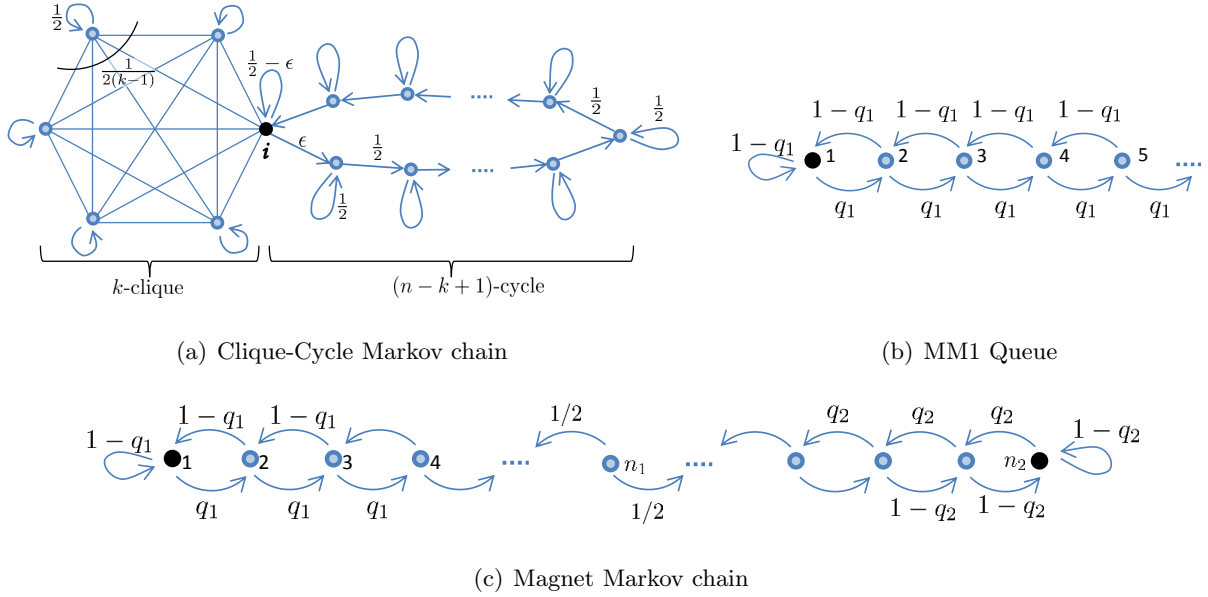
### 3. Problem Statement

Consider a discrete time, irreducible, positive recurrent Markov chain  $\{X_t\}_{t \geq 0}$  on a countable state space  $\Sigma$  with transition probability matrix  $P : \Sigma \times \Sigma \rightarrow [0, 1]$ . Given node  $i$  and threshold  $\Delta$ , is  $\pi_i > \Delta$ ? If so, what is  $\pi_i$ ? We provide a local approximation algorithm for  $\pi_i$ , which uses only edges within a local neighborhood around  $i$  of constant size with respect to the state space. We also extend this local algorithm to give approximations for the stationary probabilities of a set of nodes  $J \subseteq \Sigma$ .

DEFINITION 1. An algorithm is *local* if it only uses information within a local neighborhood of size  $r$  around  $i$ , where  $r$  is constant with respect to the size of the state space.

We illustrate the limitations of using a local algorithm for answering this question. Consider the Clique-Cycle Markov chain shown in Figure 1(a) with  $n$  nodes, composed of a size  $k$  clique connected to a size  $(n - k + 1)$  cycle. For node  $j$  in the clique excluding  $i$ , with probability  $1/2$ , the random walk stays at node  $j$ , and with probability  $1/2$  the random walk chooses a random neighbor uniformly. For node  $j$  in the cycle, with probability  $1/2$ , the random walk stays at node  $j$ , and with probability  $1/2$  the random walk travels counterclockwise to the subsequent node in the cycle. For node  $i$ , with probability  $\epsilon$  the random walk enters the cycle, with probability  $1/2$





**Figure 1** These figures depict examples of Markov chains which show the limitations of local algorithms.

the random walk chooses any neighbor in the clique; and with probability  $1/2 - \epsilon$  the random walk stays at node  $i$ . We can show that the expected return time to node  $i$  is

$$\begin{aligned} \mathbb{E}_i[T_i] &= \left(\frac{1}{2} - \epsilon\right) + \epsilon \mathbb{E}[T_i | \text{first step enters cycle}] + \frac{1}{2} \mathbb{E}[T_i | \text{first step enters clique}] \\ &= 1 + \epsilon(2(n-k)) + \frac{1}{2}(2(k-1)) = (1-2\epsilon)k + 2\epsilon n. \end{aligned} \quad (5)$$

Therefore,  $\mathbb{E}_i[T_i]$  scales linearly in  $n$  and  $k$ . Suppose we observe only the local neighborhood of constant size  $r$  around node  $i$ . All Clique-Cycle Markov chains with more than  $k + 2r$  nodes have identical local neighborhoods. Therefore, for any  $\Delta \in (0, 1)$ , there exists two Clique-Cycle Markov chains which have the same  $\epsilon$  and  $k$ , but two different values for  $n$ , such that even though their local neighborhoods are identical,  $\pi_i > \Delta$  in the Markov chain with a smaller  $n$ , while  $\pi_i < \Delta$  in the Markov chain with a larger  $n$ . Therefore, by restricting ourselves to a local neighborhood around  $i$  of constant size, we will not be able to correctly determine whether  $\pi_i > \Delta$  for every node  $i$  in any arbitrary Markov chain.

As a second example, consider the Markov chains shown in Figures 1(b) and 1(c). The state space of the Markov chain in Figure 1(b) is the positive integers (therefore countably infinite). It models the length of a MM1 queue, where  $q_1$  is the probability that an arrival occurs before a departure. This is also equivalent to a biased random walk on  $\mathbb{Z}_+$ . In the Markov chain depicted by Figure 1(c), when  $q_1$  and  $q_2$  are less than one half, nodes 1 to  $n_1 - 1$  are attracted to node 1, and nodes  $n_1$  to  $n_2$  are attracted to state  $n_2$ . Since there are two opposite attracting states (1 and  $n_2$ ), we call this Markov chain a ‘‘Magnet’’.

Consider the problem of determining if  $\pi_1$  is greater than  $\Delta$  or less than  $\Delta$ . The local neighborhoods of constant size  $r$  around node 1 for both the MM1 queue and the Magnet Markov chain are identical when  $n_1 > r$ . Therefore, there exists a Magnet Markov chain where  $n_1 > r$ , and  $q_2 \ll q_1$  such that  $\pi_1 < \Delta$  in the Magnet Markov chain, yet in the corresponding MM1 queue,  $\pi_1 > \Delta$ . Therefore, by restricting to information within a local neighborhood of node 1, it is impossible to always correctly determine if  $\pi_1 > \Delta$  or  $\pi_1 < \Delta$ .

This illustrates an unavoidable challenge for any local algorithm, including ours. For high importance nodes such that  $\pi_i > \Delta$ , our algorithm will answer correctly with high probability. However, for low importance nodes such that  $\pi_i < \Delta$ , our algorithm may still conclude that the node is important if the node “looks important” within its local neighborhood. This is unavoidable since we are trying to estimate a global property of the Markov chain using only local information<sup>8</sup>. As alluded to in the preliminary section, we will use the hitting time  $H_i$  and the fundamental matrix  $Z$  to characterize the worst case error as a function of the given Markov chain. In both the Clique-Cycle and Magnet Markov chains,  $H_i$  and entries in  $Z$  are large.

#### 4. Algorithm

Given a threshold  $\Delta \in (0, 1)$  and a node  $i \in \Sigma$ , which we will call the “anchor node”, the algorithm obtains an estimate  $\hat{\pi}_i$  of  $\pi_i$ . The algorithm relies on the characterization of  $\pi_i$  given in Lemma 1(b):  $\pi_i = 1/\mathbb{E}_i[T_i]$ . If  $\hat{\pi}_i < \Delta/(1 + \epsilon)$ , it outputs 0 (indicating  $\pi_i \leq \Delta$ ), and otherwise it outputs 1 (indicating  $\pi_i > \Delta$ ). By definition,  $\mathbb{E}_i[T_i] = \mathbb{E}[\inf\{t \geq 1 : X_t = i\} | X_0 = i]$ . In words, it is the expected time for a random walk that begins at node  $i$  to return to node  $i$ . Since the length of  $T_i$  is possibly unbounded, the algorithm truncates each sample of  $T_i$  at some predetermined maximum length, which we denote as  $\theta$ . Therefore, the algorithm generates many independent samples of a truncated random walk that begins at node  $i$  and stops either when the random walk returns to node  $i$ , or when the length exceeds  $\theta$ . Each sample is generated by simulating the random walk using “crawl” operations over the Markov chain graph  $G$ . The expected length of each sample random walk is  $\mathbb{E}_i[\min(T_i, \theta)]$ , which is close to  $\mathbb{E}_i[T_i]$  when  $\theta$  is large.

As the number of samples and  $\theta$  go to infinity, the estimate will converge almost surely to  $\pi_i$ , due to the strong law of large numbers and positive recurrence of the Markov chain. The number of samples must be large enough to guarantee that the sample mean concentrates around the true mean of the random variable. We use Chernoff’s bound (see Appendix 11) to choose a sufficiently large number of samples to guarantee that with probability  $1 - \alpha$ , the average length of the sample random walks will lie within  $(1 \pm \epsilon)$  of  $\mathbb{E}_i[\min(T_i, \theta)]$ .

We also need to choose a suitable value for  $\theta$  that balances between accuracy and computation cost. If  $\theta$  is too small, then most of the samples may be truncated, and the sample mean will be

far from the true mean; however, choosing a large  $\theta$  may cause computation to be unnecessarily expensive, since we may need a larger number of samples to guarantee concentration.

The algorithm searches iteratively for an appropriate size for the local neighborhood by beginning with a small size and increasing the size geometrically. The iterations are denoted by  $t \in \mathbb{Z}_+$ . The maximum sample path length, also equivalent to the local neighborhood size, is denoted with  $\theta^{(t)}$  and chosen to be  $2^t$ . The number of samples  $N^{(t)}$  for each iteration is determined as a function of  $\theta^{(t)}$  according to the Chernoff's bound. In our analysis, we will show that the total computation summed over all iterations is only a constant factor more than the computation in the final iteration.

**Input:** Anchor node  $i \in \Sigma$  and parameters  $\Delta =$  threshold for importance,  $\epsilon =$  closeness of the estimate, and  $\alpha =$  probability of failure.

**Initialize:** Set

$$t = 1, \theta^{(1)} = 2, N^{(1)} = \left\lceil \frac{6(1 + \epsilon) \ln(8/\alpha)}{\epsilon^2} \right\rceil.$$

**Step 1 (Gather Samples)** For each  $k$  in  $\{1, 2, 3, \dots, N^{(t)}\}$ , generate independent samples  $s_k \sim \min(T_i, \theta^{(t)})$  by simulating sample paths of the Markov chain  $X_t$  beginning at node  $i$ , and setting  $s_k$  to be the length of the  $k^{\text{th}}$  sample path. Let  $\hat{p}^{(t)} \triangleq$  the fraction of samples truncated at  $\theta^{(t)}$ , and let

$$\hat{T}_i^{(t)} \triangleq \frac{1}{N^{(t)}} \sum_{k=1}^{N^{(t)}} s_k, \quad \text{and} \quad \hat{\pi}_i^{(t)} \triangleq \frac{1}{\hat{T}_i^{(t)}}.$$

**Step 2 (Termination Conditions)**

- If **(a)**  $\hat{\pi}_i^{(t)} < \frac{\Delta}{(1+\epsilon)}$ , then stop and return **0**, and  $\hat{\pi}_i^{(t)}$ .
- Else if **(b)**  $\hat{p}^{(t)} \cdot \hat{\pi}_i^{(t)} < \epsilon \Delta$ , then stop and return **1**, and  $\hat{\pi}_i^{(t)}$ .
- Else continue.

**Step 3 (Update Rules)** Set

$$\theta^{(t+1)} \leftarrow 2 \cdot \theta^{(t)}, N^{(t+1)} \leftarrow \left\lceil \frac{3(1 + \epsilon)\theta^{(t+1)} \ln(4\theta^{(t+1)}/\alpha)}{\hat{T}_i^{(t)} \epsilon^2} \right\rceil, \text{ and } t \leftarrow t + 1.$$

Return to **Step 1**.

**Output:** **0** or **1** indicating whether  $\pi_i > \Delta$ , and estimate  $\hat{\pi}_i^{(t)}$ .

This algorithm outputs an estimate  $\hat{\pi}_i$  for the node  $i$ , in addition to **0** or **1** indicating whether  $\pi_i > \Delta$ . Since  $N^{(t)}$  is chosen to guarantee that with high probability  $\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]$  for all  $t$ , the estimate  $\hat{\pi}_i^{(t)}$  is larger than  $\pi_i/(1 + \epsilon)$  in all iterations with high probability.  $\mathbb{E}_i[\hat{T}_i^{(t)}] =$

$\mathbb{E}_i[\min(T_i, \theta^{(t)})]$  increases with each iteration; thus the expected estimate decreases in each iteration, converging to  $\pi_i$  from above as  $N^{(t)}$  and  $\theta^{(t)}$  increase.

When the algorithm terminates at stopping condition (a) and outputs  $\mathbf{0}$ , the estimate is smaller than  $\Delta/(1 + \epsilon)$ , which implies  $\pi_i < \Delta$  with high probability. Therefore, for a node  $i$  such that  $\pi_i > \Delta$ , with high probability, the algorithm will terminate at condition (b) and output  $\mathbf{1}$ . When the algorithm terminates at condition (b), the fraction of samples truncated is small, and we prove that the percentage error of estimate  $\hat{\pi}_i^{(t)}$  is upper bounded by  $O(\epsilon Z_{\max}(i))$ . When  $Z_{\max}(i)$  is large, such as for the Magnet Markov chain in Figure 1(c), there may be nodes which look important within the local neighborhood yet are not important globally, in which case our algorithm may still terminate at condition (b) and output  $\mathbf{1}$ , though  $\pi_i$  may still be less than  $\Delta$ .

We will refer to the total number of iterations used in the algorithm as the value for  $t$  at the time of termination, which we denote by  $t_{\max}$ . The total number of random walk steps taken within the first  $t$  iterations is  $\sum_{k=1}^t N^{(k)} \cdot \hat{T}_i^{(k)}$ . The algorithm always terminates within  $\ln(1/\epsilon\Delta)$  iterations. This is proved by observing that  $\hat{T}_i^{(k)} > \hat{p}^{(k)}\theta^{(k)}$ . When  $\theta^{(k)} > 1/\epsilon\Delta$ , then  $\hat{p}^{(k)}/\hat{T}_i^{(k)} < \epsilon\Delta$ , and thus termination condition (b) is satisfied. Since  $\theta^{(k)}$  governs the radius of the local neighborhood that the algorithm utilizes, this proves that our algorithm is local, since the maximum distance  $\theta^{(k)}$  is strictly upper bounded by  $1/\epsilon\Delta$ , regardless of the specific Markov chain. We will also prove that  $\mathbb{P}(T_i > \theta^{(t)})$  decays exponentially in  $\theta^{(t)}$ , and thus the algorithm will terminate sooner depending how quickly  $\mathbb{P}(T_i > \theta^{(t)})$  decays.

#### 4.1. Algorithm extension - multiple nodes

As alluded to previously, we can also simultaneously learn about other nodes in the Markov chain through these random walks from  $i$ . We refer to node  $i$  as the *anchor node*. We will extend our algorithm to obtain estimates for the stationary probability of nodes within a subset  $J \subseteq \Sigma$ , which is given as an input to the algorithm. We refer to  $J$  as the set of *observer nodes*. We estimate the stationary probability of any node  $j \in J$  using Lemma 1(a), which states that  $\pi_j$  is equivalent to the fraction of expected visits to node  $j$  out of the total number of steps taken along a random walk beginning at node  $i$  and returning to node  $i$ . The algorithm can be modified to keep track of how many times each node in  $J$  is visited along the sample paths. The estimate  $\tilde{\pi}_j$  is the fraction of visits to node  $j$  along the sampled paths. We replace Step 1 of the algorithm with Step 1\*, and when the algorithm terminates, it also outputs the values for  $\tilde{\pi}_j$  for all  $j \in J$  in addition to  $\hat{\pi}_i$ .

**Step 1\* (Gather Samples)** For each  $k$  in  $\{1, 2, 3, \dots, N^{(t)}\}$ , generate an independent sample path of the Markov chain  $\{X_0, X_1, X_2, \dots\}$  beginning at  $X_0 = i$ . Let

$$s_k = \min(T_i, \theta^{(t)}), \quad \hat{T}_i^{(t)} = \frac{1}{N^{(t)}} \sum_{k=1}^{N^{(t)}} s_k, \quad \text{and} \quad \hat{\pi}_i^{(t)} = \frac{1}{\hat{T}_i^{(t)}}.$$

Let  $\hat{p}^{(t)}$  = fraction of samples that were truncated at  $\theta^{(t)}$ . For each  $j \in J$ , let

$$f_k(j) = \sum_{r=1}^{\theta^{(t)}} \mathbb{1}\{X_r = j\} \mathbb{1}\{r \leq T_i\}, \quad \hat{F}_j^{(t)} = \frac{1}{N^{(t)}} \sum_{k=1}^{N^{(t)}} f_k(j), \quad \text{and} \quad \tilde{\pi}_j^{(t)} = \frac{\hat{F}_j^{(t)}}{\hat{T}_i^{(t)}}.$$

In our analysis we show that  $N^{(t)}$  is large enough to guarantee that  $\tilde{\pi}_j^{(t)}$  is an additive approximation of  $\mathbb{E}_i[\hat{F}_j^{(t)}]/\mathbb{E}_i[\min(T_i, \theta^{(t)})]$ . This algorithm may output two estimates for the anchor node  $i$ :  $\hat{\pi}_i$ , which relies on Lemma 1(b), and  $\tilde{\pi}_i$ , which relies on Lemma 1(a). If the random walks are not truncated at  $\theta^{(t)}$ , then these two estimates would be the same. However due to the truncation,  $\tilde{\pi}_i = (1 - \hat{p}^{(t)})/\hat{T}_i^{(t)}$ , since the average frequency of visits to node  $i$  along the sample path corresponds to the fraction of samples that return to node  $i$  before exceeding the length  $\theta^{(t)}$ . While  $\hat{\pi}_i$  is an upper bound of  $\pi_i$  with high probability due to its use of truncation,  $\tilde{\pi}_i$  is neither guaranteed to be an upper or lower bound of  $\pi_i$ . Precise analyses of the approximation errors  $|\tilde{\pi}_i^{(t)} - \pi_i|$  and  $|\tilde{\pi}_j^{(t)} - \pi_j|$  for  $j \in J$  are stated in Section 6.

## 4.2. Implementation

This algorithm is simple to implement and is easy to parallelize. It requires only  $O(|J|)$  space to keep track of the visits to each node in  $J$ , and a constant amount of space to keep track of the state of the random walk sample, and running totals such as  $\hat{p}^{(t)}$  and  $\hat{T}_i^{(t)}$ . For each random walk step, the computer only needs to fetch the local neighborhood of the current state, which is upper bounded by the maximum degree. Thus, at any given instance in time, the algorithm only needs to access a small neighborhood within the graph. Each sample is completely independent, thus the task can be distributed among independent machines. In the process of sampling these random paths, the sequence of states along the path does not need to be stored or processed upon.

Consider implementing this over a distributed network, where the graph consists of the processors and the communication links between them. Each random walk over this network can be implemented by a message passing protocol. The anchor node  $i$  initiates the random walk by sending a message to one of its neighbors chosen uniformly at random. Any node which receives the message forwards the message to one of its neighbors chosen uniformly at random. As the message travels over each link, it increments its internal counter. If the message ever returns to the anchor node  $i$ , then the message is no longer forwarded, and its counter provides a sample from  $\min(T_i, \theta)$ . When the counter exceeds  $\theta$ , then the message stops at the current node. After waiting for  $\theta$  time steps, the anchor node  $i$  can compute the estimate of its stationary probability within this network, taking into consideration the messages which have returned to node  $i$ . In addition, each observer

node  $j \in J$  can keep track of the number of times any of the messages are forwarded to node  $j$ . At the end of the  $\theta$  time steps, node  $i$  can broadcast the total number of steps to all nodes  $j \in J$  so that they can properly normalize to obtain final estimates for  $\pi_j$ .

## 5. Preliminary Results

In each iteration, the values of  $\hat{p}^{(t)}$ ,  $\hat{T}_i^{(t)}$ , and  $\hat{F}_j^{(t)}$ , are directly involved in the termination conditions and estimates. The following Lemmas use concentration results for sums of independent identically distributed random variables to show that these random variables will concentrate around their mean with high probability. Recall that the number of samples  $N^{(t)}$  depends on  $\hat{T}_i^{(t-1)}$ , which itself is a random variable. Therefore, even though completely new samples are generated in each iteration, in order to prove any result about iteration  $t$ , we still must consider the distribution over values of  $\hat{T}_i^{(t-1)}$  from the previous iteration. Lemma 3 shows that with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,  $\hat{T}_i^{(t)}$  is a  $(1 \pm \epsilon)$  approximation for  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ .

LEMMA 3. For every  $t \in \mathbb{Z}_+$ ,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t \left\{ \hat{T}_i^{(k)} \in (1 \pm \epsilon) \mathbb{E}_i[\hat{T}_i^{(k)}] \right\} \right) \geq 1 - \alpha.$$

*Proof of Lemma 3.* We will sketch the proof here and leave the details to the Appendix. Let  $A_t$  denote the event  $\left\{ \hat{T}_i^{(t)} \in (1 \pm \epsilon) \mathbb{E}_i[\hat{T}_i^{(t)}] \right\}$ . As discussed earlier,  $N^{(t)}$  is a random variable that depends on  $\hat{T}_i^{(t-1)}$ . However, conditioned on the event  $A_{t-1}$ , we can lower bound  $N^{(t)}$  as a function of  $\mathbb{E}_i[\hat{T}_i^{(t-1)}]$ . Then we apply Chernoff's bound for independent identically distributed bounded random variables and use the fact that  $\mathbb{E}_i[\hat{T}_i^{(t)}]$  is nondecreasing in  $t$  to show that

$$\mathbb{P}_i(A_t | A_{t-1}) \geq 1 - \frac{\alpha}{2^{t+1}} \quad \text{for all } t.$$

Since iteration  $t$  is only dependent on the outcome of previous iterations through the variable  $\hat{T}_i^{(t-1)}$ , we know that  $A_t$  is independent from  $A_k$  for  $k < t$  conditioned on  $A_{t-1}$ . Therefore,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t A_k \right) = \mathbb{P}_i(A_1) \prod_{k=2}^t \mathbb{P}_i(A_k | A_{k-1}).$$

We can combine these two insights to complete the proof. □

Lemma 4 shows that with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,  $\hat{p}^{(t)}$  lies within an additive  $\epsilon/3$  interval around  $\mathbb{P}(T_i > \theta^{(t)})$ . It uses similar proof techniques as Lemma 5. It is used to prove that when the algorithm terminates at stopping condition (b), with high probability,  $\mathbb{P}(T_i > \theta^{(t)}) < \epsilon(4/3 + \epsilon)$ , which is used to upper bound the estimation error.

LEMMA 4. For every  $t \in \mathbb{Z}_+$ ,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t \left\{ \hat{p}^{(k)} \in \mathbb{P}_i(T_i > \theta^{(k)}) \pm \frac{\epsilon}{3} \right\} \bigcap_{k=1}^t \left\{ \hat{T}_i^{(k)} \in (1 \pm \epsilon) \mathbb{E}_i \left[ \hat{T}_i^{(k)} \right] \right\} \right) \geq 1 - \alpha.$$

Lemma 5 shows that with probability greater than  $1 - \alpha$ , for all iterations  $t$  such that  $\mathbb{P}(T_i > \theta^{(t)}) < 1/2$ , both  $\hat{p}^{(t)}$  and  $\hat{T}_i^{(t)}$  are within  $(1 \pm \epsilon)$  of the expected values. It is used in the analysis of the estimate  $\tilde{\pi}_i^{(t)}$ . The proof is similar as above, except that we have two events per iteration to consider. Conditioning on the event that  $(1 - \hat{p}^{(t-1)})$  and  $\hat{T}_i^{(t-1)}$  are both within  $(1 \pm \epsilon)$  of their means, we compute the probability that  $(1 - \hat{p}^{(t)})$  and  $\hat{T}_i^{(t)}$  also concentrate around their means, using Chernoff's bound and union bound.

LEMMA 5. Let  $t_0$  be such that  $\mathbb{P}(T_i > \theta^{(t_0)}) < 1/2$ . For every  $t \geq t_0$ ,

$$\mathbb{P}_i \left( \bigcap_{k=t_0}^t \left\{ (1 - \hat{p}^{(k)}) \in (1 \pm \epsilon)(1 - \mathbb{P}(T_i > \theta^{(k)})) \right\} \bigcap_{k=1}^t \left\{ \hat{T}_i^{(k)} \in (1 \pm \epsilon) \mathbb{E}_i \left[ \hat{T}_i^{(k)} \right] \right\} \right) \geq 1 - \alpha.$$

Similarly, Lemma 6 shows that with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,  $\hat{F}_j^{(t)}$  and  $\hat{T}_i^{(t)}$  are close to their expected values. It is used in the analysis of the estimate  $\tilde{\pi}_j^{(t)}$  for  $j \neq i$ . Observe that  $\hat{F}_j^{(t)}$  is only guaranteed to be within an additive value of  $\epsilon \mathbb{E}_i[\hat{T}_i^{(k)}]$  around its mean. This allows us to show that the ratio between  $\hat{F}_j^{(t)}$  and  $\hat{T}_i^{(t)}$  is within an additive  $\epsilon$  error around the ratio of their respective means. We are not able to obtain a small multiplicative error bound on  $\hat{F}_j^{(t)}$  because we do not use any information from node  $j$  to choose the number of samples  $N^{(t)}$ .  $\mathbb{E}_i[\hat{F}_j^{(t)}]$  can be arbitrarily small compared to  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ , so we may not have enough samples to estimate  $\mathbb{E}_i[\hat{F}_j^{(t)}]$  closely. The remaining parts of the proof use the same techniques as described above.

LEMMA 6. For every  $t \in \mathbb{Z}_+$ ,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t \left\{ \hat{F}_j^{(k)} \in \mathbb{E}_i \left[ \hat{F}_j^{(k)} \right] \pm \epsilon \mathbb{E}_i \left[ \hat{T}_i^{(k)} \right] \right\} \bigcap_{k=1}^t \left\{ \hat{T}_i^{(k)} \in (1 \pm \epsilon) \mathbb{E}_i \left[ \hat{T}_i^{(k)} \right] \right\} \right) \geq 1 - \alpha.$$

Lemmas 3 to 6 show that  $N^{(t)}$  is large enough such that we can treat the random variables as a close approximation of their expected values. Lemmas 7 and 8 ensure that despite truncating the random walk samples at length  $\theta^{(t)}$ ,  $\mathbb{E}_i[\hat{T}_i^{(t)}]$  is still reasonably close to  $\mathbb{E}_i[T_i]$ . The truncation produces a systematic bias such that  $\hat{\pi}_i$  is larger than  $\pi_i$  with high probability. Lemma 8 shows that this bias decreases exponentially as a function of  $\theta^{(t)}$ , relying on the result from Lemma 7. Lemma 7 states that the tail of the distribution of return times to node  $i$  decays exponentially. This underlies the analysis of both the estimation error and the computation cost, or the number of iterations until one of the termination conditions are satisfied. Intuitively, it means that the distribution over return times is concentrated around its mean, since it cannot have large probability at values far away from the mean. For finite state space Markov chains, this result is easy to show using the strong Markov property, as outlined by Aldous and Fill (1999).

LEMMA 7 (**Aldous and Fill**). *Let Markov chain  $\{X_t\}$  be defined on finite state space  $\Sigma$ . For any  $i \in \Sigma$  and  $k \in \mathbb{Z}_+$ ,*

$$\mathbb{P}_i(T_i > k) \leq 2 \cdot 2^{-k/2H_i},$$

where  $H_i = \max_{j \in \Sigma} \mathbb{E}_j[T_i]$ .

Lemma 8 gives an expression for the systematic bias or difference between the true desired mean  $\mathbb{E}_i[T_i]$  and the truncated mean  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ . It shows that since  $\mathbb{P}_i(T_i > k)$  decays exponentially in  $k$ , the bias likewise decays exponentially in  $\theta^{(t)}$ .

LEMMA 8.

$$\mathbb{E}_i[T_i] - \mathbb{E}_i[\hat{T}_i^{(t)}] = \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k).$$

*Proof of Lemma 8.* Since  $T_i$  is a nonnegative random variable, and by the definition of  $\hat{T}_i^{(t)}$ ,

$$\begin{aligned} \mathbb{E}_i[T_i] - \mathbb{E}_i[\hat{T}_i^{(t)}] &= \mathbb{E}_i[T_i] - \mathbb{E}_i[\min(T_i, \theta^{(t)})] \\ &= \sum_{k=0}^{\infty} \mathbb{P}_i(T_i > k) - \sum_{k=0}^{\theta^{(t)}-1} \mathbb{P}_i(T_i > k) \\ &= \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k). \end{aligned}$$

□

## 6. Results for Estimation Error

In this section, we combine the intuition and lemmas given above into formal theorem statements about the correctness of the algorithm. The omitted proofs can be found in the Appendix. Theorem 1 states that with high probability, for any irreducible, positive recurrent Markov chain, the algorithm will correctly identify high importance nodes. Recall that the algorithm only outputs  $\mathbf{0}$  if it terminates at condition (a), which is satisfied if  $\hat{\pi}_i^{(t)} \leq \Delta/(1 + \epsilon)$ .

THEOREM 1. *For an irreducible, positive recurrent, countable state space Markov chain, and for any  $i \in \Sigma$ , with probability greater than  $(1 - \alpha)$ : If the algorithm terminates at condition (a) and outputs  $\mathbf{0}$ , then indeed  $\pi_i < \Delta$ . Equivalently, if  $\pi_i \geq \Delta$ , the algorithm will terminate at condition (b) and output  $\mathbf{1}$ .*

*Proof of Theorem 1.* This result follows from the fact that  $\hat{\pi}_i^{(t)}$  lies within  $1/(1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]$  and  $\mathbb{E}_i[\hat{T}_i^{(t)}] \leq \mathbb{E}_i[T_i]$ , thus  $\hat{\pi}_i^{(t)}$  will be an upper bound on  $\pi_i$  with high probability. By Lemma 3, with probability  $(1 - \alpha)$ , for all  $t$ ,

$$\hat{\pi}_i^{(t)} \geq \frac{\pi_i}{1 + \epsilon}.$$

Therefore, if  $\hat{\pi}_i^{(t)} < \Delta/(1 + \epsilon)$  for any  $t$ , then  $\pi_i < \Delta$  with high probability. □



Since the above theorem only relies on the property that  $\hat{\pi}_i^{(t)}$  is with high probability larger than  $\pi_i$ , when the algorithm terminates at condition (b), the theorem does not give us any guarantee of whether  $\pi_i \geq \Delta$  or  $\pi_i < \Delta$ . The following results show a tighter characterization for the estimates  $\hat{\pi}_i^{(t)}$ ,  $\tilde{\pi}_i^{(t)}$ , and  $\tilde{\pi}_j^{(t)}$  for  $j \neq i$ , as a function of the Markov chain properties. Recall that  $\hat{\pi}_i^{(t)}$  concentrates around  $1/\mathbb{E}_i[T_i^{(t)}]$ , and  $\pi_i = 1/\mathbb{E}_i[T_i]$ . Lemma 9 represents the additive difference between  $1/\mathbb{E}_i[T_i^{(t)}]$  and  $1/\mathbb{E}_i[T_i]$  as a function of  $\mathbb{P}_i(T_i > \theta^{(t)})$ ,  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ , and the fundamental matrix  $Z$ .

LEMMA 9. *For an irreducible, positive recurrent Markov chain  $\{X_t\}$  with countable state space  $\Sigma$  and transition probability matrix  $P$ , and for any  $i \in \Sigma$  and  $t \in \mathbb{Z}_+$ ,*

$$\frac{1}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_i = \frac{\mathbb{P}_i(T_i > \theta^{(t)}) \Gamma_i}{\mathbb{E}_i[\hat{T}_i^{(t)}]}, \quad (6)$$

where

$$\Gamma_i \triangleq \left( \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) (Z_{ii} - Z_{qi}) \right). \quad (7)$$

*Proof of Lemma 9.* We divide the equation given in Lemma 8 by  $\mathbb{E}_i[T_i]$  and  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ . Then we apply Bayes' rule, the law of total probability, and the Markov property.

$$\begin{aligned} \frac{1}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \frac{1}{\mathbb{E}_i[T_i]} &= \frac{1}{\mathbb{E}_i[T_i]\mathbb{E}_i[\hat{T}_i^{(t)}]} \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k) \\ &= \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{\mathbb{E}_i[T_i]\mathbb{E}_i[\hat{T}_i^{(t)}]} \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k | T_i > \theta^{(t)}) \\ &= \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{\mathbb{E}_i[T_i]} \sum_{k=\theta^{(t)}}^{\infty} \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(T_i > k | X_{\theta^{(t)}} = q, T_i > \theta^{(t)}) \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \\ &= \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \frac{\mathbb{E}_q[T_i]}{\mathbb{E}_i[T_i]}. \end{aligned} \quad (8)$$

Finally, we use Lemma 2 to complete the proof. □

In order to understand the expression  $\Gamma_i$ , we observe that by the definition of  $Z$ ,

$$Z_{ii} - Z_{qi} = \sum_{k=0}^{\infty} (P_{ii}^{(k)} - P_{qi}^{(k)}).$$

Since both  $P_{ii}^{(k)}$  and  $P_{qi}^{(k)}$  converge to  $\pi_i$  as  $k$  goes to infinity, this is similar to the notion of “mixing time”. However,  $\Gamma_i$  further weights each quantity  $(Z_{ii} - Z_{qi})$  with the distribution  $\mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)})$ . Because all of the random walks begin at node  $i$ , we expect for this distribution to more heavily weight nodes which are closer to  $i$ . Thus,  $\Gamma_i$  can be interpreted as a “local mixing time”, which measures the time it takes for a random walk beginning at some distribution of nodes around  $i$  to reach stationarity at node  $i$ .

In fact, this lemma gives us a key insight into the termination conditions of the algorithm. Recall that termination condition (b) is satisfied when  $\hat{p}^{(t)} \cdot \hat{\pi}_i^{(t)} < \epsilon \Delta$ . Since  $\hat{p}^{(t)} \cdot \hat{\pi}_i^{(t)}$  concentrates around  $\mathbb{P}_i(T_i > \theta^{(t)}) / \mathbb{E}_i[\hat{T}_i^{(t)}]$ , Lemma 9 indicates that when the algorithm stops at condition (b), the additive error between  $\hat{\pi}_i^{(t)}$  and  $\pi_i$  is approximately  $\epsilon \Delta$  times a function of the fundamental matrix  $Z$ . We can naively upper bound  $\Gamma_i$  by  $2Z_{\max}(i)$ , where  $Z_{\max}(i) = \max_k |Z_{ki}|$  for  $Z_{ki} = \sum_{t=1}^{\infty} (P_{ki}^{(t)} - \pi_i)$ . This gives us a complete picture of the performance of our algorithm. When the algorithm terminates at condition (a), we are guaranteed that  $\pi_i < \Delta$ . When the algorithm terminates at condition (b), the estimate  $\hat{\pi}_i^{(t)}$  is within an  $\epsilon \Delta$  additive factor of  $\pi_i$ , modulated by  $Z_{\max}(i)$ , which is a function of the Markov chain and node  $i$ .

Theorem 2 states that with high probability, the percentage error between  $\hat{\pi}_i^{(t)}$  and  $\pi_i$  is upper bounded by a function of  $\epsilon$ ,  $\theta^{(t)}$ ,  $Z_{\max}(i)$ , and  $H_i$ . This is proved by combining Lemmas 9, 3, 4, and 7.

**THEOREM 2.** *For an irreducible Markov chain  $\{X_t\}$  with finite state space  $\Sigma$  and transition probability matrix  $P$ , for any  $i \in \Sigma$ , with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,*

$$\begin{aligned} \left| \frac{\hat{\pi}_i^{(t)} - \pi_i}{\hat{\pi}_i^{(t)}} \right| &\leq 2(1 - \epsilon) \mathbb{P}_i(T_i > \theta^{(t)}) Z_{\max}(i) + \epsilon, \\ &\leq 4(1 - \epsilon) 2^{-\theta^{(t)}/2H_i} Z_{\max}(i) + \epsilon. \end{aligned}$$

Therefore, with probability greater than  $1 - \alpha$ , if the algorithm terminates at condition (b), then

$$\left| \frac{\hat{\pi}_i^{(t_{\max})} - \pi_i}{\hat{\pi}_i^{(t_{\max})}} \right| \leq \epsilon (3Z_{\max}(i) + 1).$$

The error bound decays exponentially in  $\theta^{(t)}$ , which doubles in each iteration. Thus, for every subsequent iteration  $t$ , the estimate  $\hat{\pi}_i^{(t)}$  approaches  $\pi_i$  exponentially fast. This relies on the fact that the distribution of the return time  $T_i$  has an exponentially decaying tail, ensuring that the return time  $T_i$  concentrates around its mean  $\mathbb{E}_i[T_i]$ . When the algorithm terminates at stopping condition (b),  $\mathbb{P}(T_i > \theta) \leq \epsilon(4/3 + \epsilon)$  with high probability, thus the percentage error is bounded by  $O(\epsilon Z_{\max}(i))$ .

Next we analyze  $\tilde{\pi}_i^{(t)}$ , which we recall is equal to  $(1 - \hat{p}^{(t)}) / \hat{T}_i^{(t)}$ . Lemma 10 gives an expression for the additive difference between  $(1 - \mathbb{P}_i(T_i > \theta^{(t)})) / \mathbb{E}_i[T_i^{(t)}]$  and  $1 / \mathbb{E}_i[T_i]$  as a function of  $\mathbb{P}_i(T_i > \theta^{(t)})$ ,  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ , and the fundamental matrix  $Z$ .

**LEMMA 10.** *For an irreducible, positive recurrent Markov chain  $\{X_t\}$  with countable state space  $\Sigma$  and transition probability matrix  $P$ , and for any  $i \in \Sigma$  and  $t \in \mathbb{Z}_+$ ,*

$$\frac{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_i = \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} (\Gamma_i - 1). \quad (10)$$

*Proof of Lemma 10.* This lemma follows directly from Lemma 9.  $\square$

In comparing Lemma 9 and 10, we see that the estimation errors are almost the same except for a minus one in Lemma 10. Therefore, when  $\Gamma_i$  is small, we expect  $\tilde{\pi}_i$  to be a better estimate than  $\hat{\pi}_i$ ; however, when  $\Gamma_i$  is large, then the two estimates will have approximately the same error. Theorem 3 combines Lemma 10 with the concentration results from Lemma 5 to show that with high probability, the percentage error between  $\tilde{\pi}_i^{(t)}$  and  $\pi_i$  decays exponentially in  $\theta^{(t)}$ . We require  $\mathbb{P}_i(T_i > \theta^{(t)}) < 1/2$  in order to ensure that  $(1 - \hat{p}^{(t)})$  concentrates within a  $(1 \pm \epsilon)$  multiplicative interval around  $(1 - \mathbb{P}_i(T_i > \theta^{(t)}))$ . We will show simulations of computing PageRank, in which  $\tilde{\pi}_i$  is a closer estimate of  $\pi_i$  than  $\hat{\pi}_i$ .

**THEOREM 3.** *For an irreducible Markov chain  $\{X_t\}$  with finite state space  $\Sigma$  and transition probability matrix  $P$ , for any  $i \in \Sigma$ , with probability greater than  $1 - \alpha$ , for all iterations  $t$  such that  $\mathbb{P}(T_i > \theta^{(t)}) < 1/2$ ,*

$$\begin{aligned} \left| \frac{\tilde{\pi}_i^{(t)} - \pi_i}{\tilde{\pi}_i^{(t)}} \right| &\leq \left( \frac{1 + \epsilon}{1 - \epsilon} \right) \left( \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{1 - \mathbb{P}_i(T_i > \theta^{(t)})} \right) \max(2Z_{\max}(i) - 1, 1) + \frac{2\epsilon}{1 - \epsilon}, \\ &\leq \frac{4(1 + \epsilon)}{1 - \epsilon} 2^{-\theta^{(t)}/2H_i} \max(2Z_{\max}(i) - 1, 1) + \frac{2\epsilon}{1 - \epsilon}. \end{aligned}$$

Finally, we analyze  $\tilde{\pi}_j^{(t)} = \hat{F}_j^{(t)}/\hat{T}_i^{(t)}$  for  $j \neq i$ . Lemma 11 gives an expression for the additive difference between  $\mathbb{E}_i[F_j^{(t)}]/\mathbb{E}_i[T_i^{(t)}]$  and  $\mathbb{E}_i[F_j]/\mathbb{E}_i[T_i]$  as a function of  $\mathbb{P}_i(T_i > \theta^{(t)})$ ,  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ , and the fundamental matrix  $Z$ .

**LEMMA 11.** *For an irreducible, positive recurrent Markov chain  $\{X_t\}$  with countable state space  $\Sigma$  and transition probability matrix  $P$ , and for any  $i, j \in \Sigma$ , and  $t \in \mathbb{Z}_+$ ,*

$$\frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_j = \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \left( \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) (Z_{ij} - Z_{qj}) \right) \quad (11)$$

Compare Lemmas 9 and 11. Although they look similar, observe that if we naively bound  $(Z_{ij} - Z_{qj})$  by  $2Z_{\max}(j)$ , it becomes clear that Lemma 11 depends on the Markov chain mixing properties with respect to both node  $i$  through  $\mathbb{P}_i(T_i > \theta^{(t)})$ , and node  $j$  through  $Z_{\max}(j)$ . Theorem 4 bounds the error for the estimates  $\tilde{\pi}_j^{(t)}$  for nodes  $j \neq i$  by combining the concentration results in Lemma 6 with Lemma 11. Due to the looser additive concentration guarantees for  $\hat{F}_i^{(t)}$ , Theorem 4 provides an additive error bound rather than a bound on the percentage error.

**THEOREM 4.** *For an irreducible Markov chain  $\{X_t\}$  with finite state space  $\Sigma$  and transition probability matrix  $P$ , for any  $i, j \in \Sigma$  such that  $j \neq i$ , with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,*

$$\begin{aligned} \left| \tilde{\pi}_j^{(t)} - \pi_j \right| &\leq 2(1 + \epsilon) \mathbb{P}_i(T_i > \theta^{(t)}) Z_{\max}(j) \hat{\pi}_i^{(t)} + \epsilon \tilde{\pi}_j^{(t)} + \epsilon, \\ &\leq 4(1 + \epsilon) 2^{-\theta^{(t)}/2H_i} Z_{\max}(j) \hat{\pi}_i^{(t)} + \epsilon \tilde{\pi}_j^{(t)} + \epsilon. \end{aligned}$$

The first term in the expression decays exponentially in  $\theta^{(t)}$  and scales with  $\hat{\pi}_i^{(t)}$ . Instead of a dependence on  $Z_{\max}(i)$ , the expression depends on  $Z_{\max}(j)$ . In order for  $\tilde{\pi}_j$  to be a good estimate of  $\pi_j$ , both the anchor node  $i$  and the observer node  $j$  must have reasonable mixing and connectivity properties within the Markov chain.

*Tightness of Analysis* In this section, we discuss the tightness of our analysis. Lemmas 9, 10, and 11 give exact expressions of the estimation error that arises from the truncation of the sample random walks. For a specific Markov chain, Theorems 2, 3, and 4 could be loose due to two approximations. First,  $2Z_{\max}(i)$  could be a loose upper bound upon  $\Gamma_i$ . Second, Lemma 7 could be loose due to its use of the Markov inequality. Since  $\hat{\pi}_i$  is greater than  $\pi_i$  with high probability, Theorem 2 is only useful when the upper bound is less than 1. We will show that for a specific family of graphs, namely clique graphs, our bound scales correctly as a function of  $\theta^{(t)}$ ,  $H_i$ , and  $Z_{\max}(i)$ .

Consider a family of clique graphs  $G_n$  indexed by  $n \in \mathbb{Z}_+$ , such that  $G_n$  is the clique graph over  $n$  vertices. We compute the hitting time  $H_i$ , the fundamental matrix  $Z$ , and the truncation probability  $\mathbb{P}_i(T_i > k)$  for  $G_n$ , and substitute these into Lemma 9 and Theorem 2 to compare the expressions. Lemma 9 states that

$$1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} = e^{-(\theta^{(t)}-1)/(n-1)} \left( \frac{n-1}{n} \right).$$

Theorem 2 states that with probability at least  $1 - \alpha$ ,

$$\left| 1 - \frac{\pi_i}{\hat{\pi}_i^{(t)}} \right| \leq 4(1 - \epsilon)e^{-\theta^{(t)} \ln(2)/2n} \left( \frac{n^2 - n + 1}{n^2} \right) + \epsilon.$$

While Theorem 2 gives that the percentage error is upper bounded by  $O(e^{-\theta^{(t)} \ln(2)/2H_i} Z_{\max}(i))$ , by Lemma 9 the percentage error of our algorithm on the clique graph is no better than  $\Omega(e^{-\theta^{(t)}/H_i} Z_{\max}(i))$ . Consider using these bounds to determine how large the threshold  $\theta^{(t)}$  needs to be in order to guarantee that our approximation is within a small margin of error. The required threshold computed by Theorem 2 would only be a constant factor of  $2/\ln(2)$  larger than the threshold computed using Lemma 9. Our algorithm leverages the property that there is some concentration of measure, or “locality”, over the state space. It is the worst when there is no concentration of measure, and the random walks spread over the graph quickly and take a long time to return, such as in the clique graph. In the example of graphs that have stronger concentration of measure, such as biased random walk on the positive integers, the techniques presented in Section 8 using Lyapunov functions will obtain tighter bounds, as compared to using just the hitting time  $H_i$  and  $Z_{\max}(i)$ , since these quantities are computed by taking max over all states, even if the random walk may only stay within a local region around  $i$ .

## 7. Results for Computation Time

In this section, we show that the algorithm always terminates. Using the exponential tail bound in Lemma 7, we prove that if  $\theta^{(t)}$  is large enough, one of the two termination conditions will hold. In order to analyze the total computation time of the algorithm given the maximum number of iterations, we prove that the total number of random walk steps taken by the algorithm within the first  $t$  iterations scales with  $2^t$ , which we recall is equivalent to  $\theta^{(t)}$  by design.

LEMMA 12. *With probability greater than  $(1 - \alpha)$ , the total number of random walk steps taken by the algorithm within the first  $t$  iterations is bounded by*

$$\tilde{O}\left(\frac{\ln(\frac{1}{\alpha})2^t}{\epsilon^2}\right).$$

*Proof of Lemma 12.* The total number of random walk steps (i.e., neighbor queries of the graph) used in the algorithm over all  $t$  iterations is equal to  $\sum_{k=1}^t N^{(k)}\hat{T}_i^{(k)}$ . We condition on the event that  $\hat{T}_i^{(k)}$  is within a  $(1 \pm \epsilon)$  multiplicative interval around its mean, which occurs with probability greater than  $(1 - \alpha)$  by Lemma 3. Because  $\theta^{(k)}$  doubles in each iteration,  $\mathbb{E}_i[\hat{T}_i^{(k)}] \leq 2\mathbb{E}_i[\hat{T}_i^{(k-1)}]$ . By combining these facts with the definition of  $N^{(k)}$ , we obtain an upper bound as a function of  $\theta^{(k)}$ ,  $\alpha$ ,  $\epsilon$ , and  $\mathbb{E}_i[\hat{T}_i^{(k)}]$ . We suppress the insignificant factors  $(1 + \epsilon)$  and  $(1 - \epsilon)$ . Since  $N^{(k)}\hat{T}_i^{(k)}$  grows super exponentially, the largest term of the summation dominates.

$$\begin{aligned} \sum_{k=1}^t N^{(k)}\hat{T}_i^{(k)} &= \sum_{k=1}^t O\left(\frac{2^k \ln(2^k/\alpha)}{\epsilon^2}\right) \\ &= O\left(\frac{2^t \ln(2^t/\alpha)}{\epsilon^2}\right) = \tilde{O}\left(\frac{2^t \ln(\frac{1}{\alpha})}{\epsilon^2}\right). \end{aligned}$$

□

Theorem 5 asserts that with high probability, the algorithm terminates in finite time as a function of the parameters of the algorithm, independent from the size of the Markov chain state space. Therefore this implies that our algorithm is local. It is proved by showing that if  $\theta^{(t)} > 1/\epsilon\Delta$ , then termination condition (b) of the algorithm will always be satisfied.

THEOREM 5. *For an irreducible, positive recurrent, countable state space Markov chain, and for any  $i \in \Sigma$ , with probability 1, the total number of iterations used by the algorithm with parameters  $\Delta, \epsilon$ , and  $\alpha$  is bounded above by*

$$t_{\max} \leq \ln\left(\frac{1}{\epsilon\Delta}\right).$$

*With probability greater than  $1 - \alpha$ , the computation time, or the total number of steps (i.e. neighbor queries) used by the algorithm is bounded above by*

$$\sum_{k=1}^{t_{\max}} N^{(k)} \cdot \hat{T}_i^{(k)} \leq \tilde{O}\left(\frac{\ln(\frac{1}{\alpha})}{\epsilon^3\Delta}\right).$$

*Proof of Theorem 5.* By definition,  $\hat{T}_i^{(t)} \geq \hat{p}^{(t)}\theta^{(t)}$ , which implies that  $\hat{p}^{(t)}\hat{\pi}_i^{(t)} \leq 1/\theta^{(t)}$ . Recall that termination condition (b) is satisfied for  $\hat{p}^{(t)}\hat{\pi}_i^{(t)} < \epsilon\Delta$ . Therefore, when  $\theta^{(t)} \geq 1/\epsilon\Delta$ , termination condition (b) will always be satisfied. This provides an upper bound for  $t_{\max}$ , and we can substitute it into Lemma 12 to complete the proof.  $\square$

As can be seen through the proof, Theorem 5 does not utilize any information or properties of the Markov chain. On the other hand, Lemma 7 shows that depending on the properties of the Markov chain, we expect the return time to concentrate more sharply around the mean. Termination condition (a) is satisfied when  $\hat{\pi}_i^{(t)} < \Delta/(1+\epsilon)$ . We prove part (a) of Theorem 6 is by showing that for nodes  $i$  such that  $\pi_i < (1-\epsilon)\Delta/(1+\epsilon)$ , since  $\hat{T}_i^{(t)} - \mathbb{E}_i[T_i]$  decreases exponentially with  $\theta^{(t)}$ , for large enough  $t$ , termination condition (a) is satisfied with high probability. Termination condition (b) is satisfied when  $\hat{p}^{(t)} \cdot \hat{\pi}_i^{(t)} < \epsilon\Delta$ . We prove part (b) of Theorem 6 is by showing that since  $\mathbb{P}_i(T_i > \theta^{(t)})$  decreases exponentially with  $\theta^{(t)}$ , for large  $t$ , termination condition (b) will be satisfied with high probability.

**THEOREM 6.** *For an irreducible Markov chain  $\{X_t\}$  with finite state space  $\Sigma$  and transition probability matrix  $P$ ,*

**(a)** *For any node  $i \in \Sigma$  such that  $\pi_i < (1-\epsilon)\Delta/(1+\epsilon)$ , with probability greater than  $1-\alpha$ , the total number of steps used by the algorithm is bounded above by*

$$\sum_{k=1}^{t_{\max}} N^{(k)} \cdot \hat{T}_i^{(k)} \leq \tilde{O} \left( \frac{\ln(\frac{1}{\alpha})}{\epsilon^2} \left( H_i \ln \left( \left( \frac{1}{1-2^{-1/2H_i}} \right) \left( \frac{1}{\pi_i} - \frac{1+\epsilon}{(1-\epsilon)\Delta} \right)^{-1} \right) \right) \right).$$

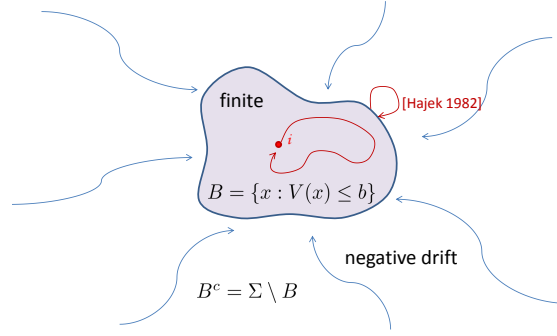
**(b)** *For all nodes  $i \in \Sigma$ , with probability greater than  $1-\alpha$ , the total number of steps used by the algorithm is bounded above by*

$$\sum_{k=1}^{t_{\max}} N^{(k)} \cdot \hat{T}_i^{(k)} \leq \tilde{O} \left( \frac{\ln(\frac{1}{\alpha})}{\epsilon^2} \left( \frac{H_i}{\alpha} \ln \left( \pi_i \left( \frac{1}{\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right) \right) \right) \right).$$

For nodes  $i$  in a Markov chain such that the maximal hitting time is small, the bounds given in Theorem 6 will be smaller than the general bound given in Theorem 5. Given a node  $i$  such that  $\pi_i < (1-\epsilon)\Delta/(1+\epsilon)$ , our tightest bound is given by the minimum over the expressions from Theorems 5, 6(a), and 6(b). For a node  $i$  such that  $\pi_i > \Delta$ , Theorem 6(a) does not apply since the algorithm will never satisfy termination condition (a) with high probability, as discussed in Theorem 1. Therefore, our tightest bound is given by the minimum between Theorem 5 and 6(b).

## 8. Results for Countable State Space Markov Chains

Theorems 2, 3, 4, and 6 only apply for finite state space Markov chains due to their dependence on Lemma 7. It relies on the finite size of the state space in order to prove the exponential decaying tail of the return time probabilities. These results can be extended to countably infinite state space



**Figure 2** This illustrates the implication of Assumption 1, which uses a Lyapunov function to decompose the state space into a finite region  $B$  and a region with negative drift.

Markov chains under some assumptions. We use Lyapunov analysis techniques to prove that the tail of the distribution of  $T_i$  decays exponentially for any node  $i$  in any countable state space Markov chain that satisfies Assumption 1.

ASSUMPTION 1. *The Markov chain  $\{X_t\}$  is irreducible. There exists a Lyapunov function  $V : \Sigma \rightarrow \mathbb{R}_+$  and constants  $\nu_{\max}, \gamma > 0$ , and  $b \geq 0$ , that satisfy the following conditions:*

1. *The set  $B = \{x \in \Sigma : V(x) \leq b\}$  is finite,*
2. *For all  $x, y \in \Sigma$  such that  $\mathbb{P}(X_{t+1} = y | X_t = x) > 0$ ,  $|V(y) - V(x)| \leq \nu_{\max}$ ,*
3. *For all  $x \in \Sigma$  such that  $V(x) > b$ ,  $\mathbb{E}[V(X_{t+1}) - V(X_t) | X_t = x] < -\gamma$ .*

At first glance, this assumption may seem very restrictive. But in fact, this is quite reasonable: by the Foster-Lyapunov criteria (cf. Theorem 12), a countable state space Markov chain is positive recurrent if and only if there exists a Lyapunov function  $V : \Sigma \rightarrow \mathbb{R}_+$  that satisfies condition (1) and (3), as well as (2'):  $\mathbb{E}[V(X_{t+1}) | X_t = x] < \infty$  for all  $x \in \Sigma$ . Assumption 1 has (2), which is a restriction of the condition (2'). The implications of Assumption 1 are visualized in Figure 2. The existence of the Lyapunov function allows us to decompose the state space into sets  $B$  and  $B^c$  such that for all nodes  $x \in B^c$ , there is an expected decrease in the Lyapunov function in the next step or transition. Therefore, for all nodes in  $B^c$ , there is a negative drift towards set  $B$ . In addition, in any single step, the random walk cannot escape “too far”. The Lyapunov function helps to impose a natural ordering over the state space that allows us to prove properties of the Markov chain. There have been many results that use Lyapunov analysis to give bounds on the stationary probabilities, return times, and distribution of return times as a function of the Lyapunov function (Hajek 1982, Bertsimas et al. 1998). Building upon results by Hajek, we prove the following lemma which establishes that return times have exponentially decaying tails even for countable-state space Markov chains, as long as they satisfy Assumption 1.

LEMMA 13. Let  $\{X_t\}$  be an irreducible Markov chain satisfying Assumption 1. For any  $i \in B$  and for all  $k \in \mathbb{Z}_+$ ,

$$\mathbb{P}_i(T_i > k) \leq 4 \cdot 2^{-\frac{k}{R_i}},$$

where

$$R_i = O\left(\frac{H_i^B e^{2\eta\nu_{\max}}}{(1-\rho)(e^{\eta\nu_{\max}} - \rho)}\right),$$

and  $H_i^B$  is the maximal hitting time over the Markov chain with its state space restricted to the subset  $B$ . The scalars  $\eta$  and  $\rho$  are functions of  $\gamma$  and  $\nu_{\max}$  (see (44) in Appendix F).

Lemma 13 pertains to nodes  $i \in B$  such that  $V(i) \leq b$ . This is not actually restrictive, since for any node  $k$  of interest such that  $V(k) = b' > b$ , we can define a new Lyapunov function  $V'(\cdot)$  such that  $V'(k) = b$ , and  $V'(j) = V(j)$  for all  $j \neq k$ . Then we define  $B' = \{j \in \Sigma : V'(j) \leq b\} = B \cup \{k\}$  and  $\nu'_{\max} = \nu_{\max} + b' - b$ . By extension, Assumption 1 holds for  $V'(\cdot)$  with constants  $\nu'_{\max}$ ,  $\gamma$ , and  $b'$ .

The quantity  $R_i$  in Lemma 13 for countable state space Markov chains plays the same role as  $H_i$  in Lemma 7 for finite state space Markov chains. Thus, the equivalent theorems for the countable state space setting are obtained by using Lemma 13 rather than Lemma 7. In the countable state space setting,  $H_i$  and  $Z_{\max}(i)$  no longer are well defined since the maximum over an infinite set may not exist. However, we recall that  $Z_{\max}(i)$  is on the same order as  $\pi_i H_i$ . Therefore, the following theorems can be understood as a modification of the theorems for finite state space by replacing  $H_i$  with  $R_i$  and replacing  $Z_{\max}(i)$  with  $\pi_i R_i$ .

Theorems 7 to 9 are the countable state space versions of Theorems 2, 3, and 6. Theorem 4 does not directly extend because in a countably infinite state space Markov chain, node  $j$  can be arbitrarily far away from node  $i$  such that random walks beginning at node  $i$  rarely hit node  $j$  before returning to node  $i$ .

THEOREM 7. For a Markov chain satisfying Assumption 1, for any  $i \in B$ , with probability greater than  $1 - \alpha$ , for all iterations  $t$ ,

$$\left| \frac{\hat{\pi}_i^{(t)} - \pi_i}{\hat{\pi}_i^{(t)}} \right| \leq 4(1 - \epsilon) \left( \frac{2^{-\theta^{(t)}/R_i}}{1 - 2^{-1/R_i}} \right) \pi_i + \epsilon.$$

THEOREM 8. For a Markov chain satisfying Assumption 1, for any  $i \in B$ , with probability greater than  $1 - \alpha$ , for all iterations  $t$  such that  $\mathbb{P}(T_i > \theta^{(t)}) < 1/2$ ,

$$\left| \frac{\tilde{\pi}_i^{(t)} - \pi_i}{\tilde{\pi}_i^{(t)}} \right| \leq \frac{8(1 + \epsilon)}{1 - \epsilon} 2^{-\theta^{(t)}/R_i} \max\left(\frac{\pi_i}{1 - 2^{-1/R_i}}, 1\right) + \frac{2\epsilon}{1 - \epsilon}.$$



**THEOREM 9.** *For a Markov chain satisfying Assumption 1,*

**(a)** *For any node  $i \in B$  such that  $\pi_i < (1 - \epsilon)\Delta/(1 + \epsilon)$ , with probability greater than  $1 - \alpha$ , the total number of steps used by the algorithm is bounded above by*

$$\sum_{k=1}^{t_{\max}} N^{(k)} \cdot \hat{T}_i^{(k)} \leq \tilde{O} \left( \frac{\ln(\frac{1}{\alpha})}{\epsilon^2} \left( R_i \ln \left( \left( \frac{1}{1 - 2^{-1/R_i}} \right) \left( \frac{1}{\pi_i} - \frac{1 + \epsilon}{(1 - \epsilon)\Delta} \right)^{-1} \right) \right) \right).$$

**(b)** *For all nodes  $i \in B$ , with probability greater than  $1 - \alpha$ , the total number of steps used by the algorithm is bounded above by*

$$\sum_{k=1}^{t_{\max}} N^{(k)} \cdot \hat{T}_i^{(k)} \leq \tilde{O} \left( \frac{\ln(\frac{1}{\alpha})}{\epsilon^2} \left( \frac{R_i}{\alpha} \ln \left( \pi_i \left( \frac{1}{\epsilon\Delta} + \frac{1}{1 - 2^{-1/R_i}} \right) \right) \right) \right).$$

## 9. Examples and Simulations

We discuss a few applications of our algorithm to concrete examples of Markov chains. The examples illustrate the wide applicability of our local algorithm for estimating stationary probabilities.

**EXAMPLE 1 (PAGERANK).** In analyzing the web graph, PageRank is a frequently used measure to compute the importance of webpages. We are given a scalar parameter  $\beta$  and an underlying directed graph over  $n$  nodes, described by the adjacency matrix  $A$  (i.e.,  $A_{ij} = 1$  if  $(i, j) \in E$  and 0 otherwise). The transition probability matrix of the PageRank random walk is given by

$$P = \frac{\beta}{n} \mathbf{1} \cdot \mathbf{1}^T + (1 - \beta) D^{-1} A, \quad (12)$$

where  $D$  denotes the diagonal matrix whose diagonal entries are the out-degrees of the nodes. It follows that

$$P_{rs} = \mathbb{P}(X_{t+1} = s | X_t = r) = \beta \left( \frac{1}{n} \right) + (1 - \beta) \left( \frac{\mathbb{1}\{A_{rs} = 1\}}{\text{out-degree}(r)} \right).$$

Thus, in every step, there is a  $\beta$  probability of jumping uniformly randomly to any other node in the graph. In our simulation,  $\beta = 0.15$ , and the underlying graph is generated according to the configuration model with a power law degree distribution:  $\mathbb{P}(d) \propto d^{-1.5}$ .

**EXAMPLE 2 (QUEUEING SYSTEM).** In queuing theory, Markov chains are commonly used to model the length of the queue of jobs waiting to be processed by a server, which evolves over time as jobs arrive and are processed. For illustrative purposes, we chose the MM1 queue, equivalent to a random walk on  $\mathbb{Z}_+$ . The state space  $\mathbb{Z}_+$  is countably infinite. Assume we have a single server where the jobs arrive according to a Poisson process with parameter  $\lambda$ , and the processing time for a single job is distributed exponentially with parameter  $\mu$ . The queue length can be modeled with the random walk shown in Figure 1(b), where  $q_1$  is the probability that a new job arrives before the current job is finished processing, given by  $\lambda/(\lambda + \mu)$ . For the purposes of our simulation, we choose  $q_1 = 0.3$ , and estimate the stationary probabilities for the queue to have length  $i$  for  $i \in \{1, 2, 3, \dots, 50\}$ .

EXAMPLE 3 (MAGNET GRAPH). This example illustrates a Markov chain with poor mixing properties. The Markov chain is depicted in Figure 1(c), and can be described as a random walk over a finite section of the integers such that there are two attracting states, labeled in Figure 1(c) as states 1 and  $n_2$ . We assume that  $q_1, q_2 < 1/2$ , such that for all states left of state  $n_1$ , the random walk will drift towards state 1 with probability  $1 - q_1$  in each step, and for all states right of state  $n_1$ , the random walk will drift towards state  $n_2$  with probability  $1 - q_2$  in each step. Due to this bipolar attraction, a random walk that begins on the left will tend to stay on the left, and similarly, a random walk that begins on the right will tend to stay on the right. For our simulations, we chose  $q_1 = q_2 = 0.3$ ,  $n_1 = 25$ , and  $n_2 = 50$ .

### 9.1. Algorithm Results for the Anchor Node

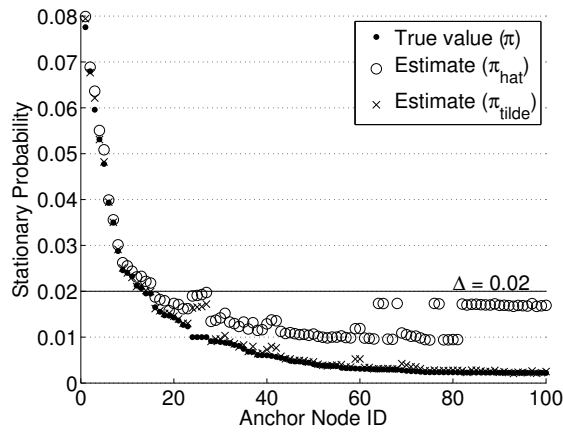
We show the results of applying our algorithm to estimate the stationary probabilities in these three different Markov chains, using parameters  $\Delta = 0.02$ ,  $\epsilon = 0.15$ , and  $\alpha = 0.2$ . Figure 3 plots the final estimates along with the true stationary probabilities. The three Markov chains have different mixing properties, chosen to illustrate the performance of our algorithm on Markov chains with different values of  $H_i$  and  $Z_{\max}(i)$ .

Figures 3(a), 3(c), and 3(e) plot the estimates  $\hat{\pi}_i^{(t_{\max})}$  and  $\tilde{\pi}_i^{(t_{\max})}$  for the anchor nodes. Thus, the data points for different anchor nodes are results from running independent instances of the algorithm with different nodes chosen as the anchor. In the PageRank Markov chain, we label the nodes in decreasing order of their true stationary probability  $\pi_i$ , since there is no natural ordering over nodes. In the MM1 queue and the Magnet Markov chains, the nodes are labeled according to Figures 1(b) and 1(c).

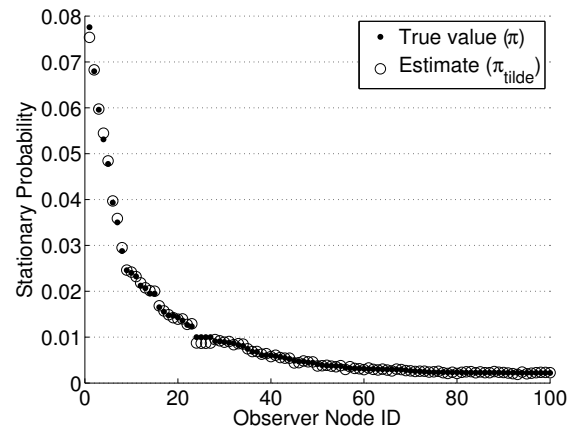
Figure 3(a) shows the result for PageRank. For nodes such that  $\pi_i > \Delta$ ,  $\hat{\pi}_i$  is a close approximation for  $\pi_i$ . For nodes such that  $\pi_i \leq \Delta$ , the algorithm successfully categorizes the node as unimportant (i.e.  $\hat{\pi}_i \leq \Delta$ ). In addition, we observe that  $\tilde{\pi}_i$  is extremely close to  $\pi_i$  for most nodes. We verify that  $(Z_{ii} - Z_{qi})$  is close to 1 for most pairs of nodes  $(i, q)$  in the PageRank Markov chain. Therefore  $\Gamma_i \approx 1$ , and Lemmas 9 and 10 indicate that  $\tilde{\pi}_i$  is expected to be a closer estimate than  $\hat{\pi}_i$ , which is consistent with our results.

Figure 3(c) plots the result for the MM1 queue Markov chain. Both estimates  $\hat{\pi}_i$  and  $\tilde{\pi}_i$  are close to  $\pi_i$ . Since the stationary probabilities decay exponentially, only the first few nodes are significant, and the algorithm correctly captures that. Both the PageRank and the MM1 queue Markov chains mix well, and as expected, our algorithm performs well.

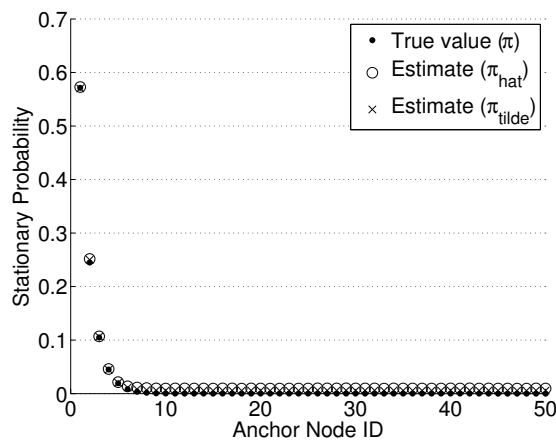
Figure 3(e) shows the result for the Magnet Markov chain, which mixes very slowly. The algorithm overestimates the stationary probabilities by almost two times the true value. This is due to



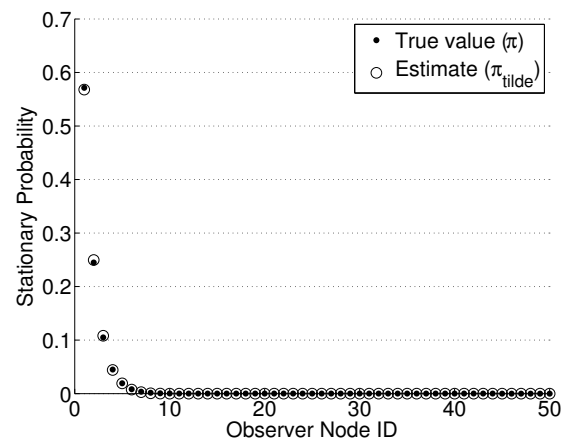
(a) PageRank.



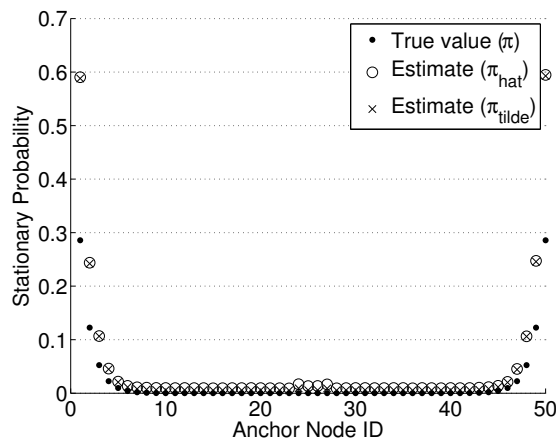
(b) PageRank.



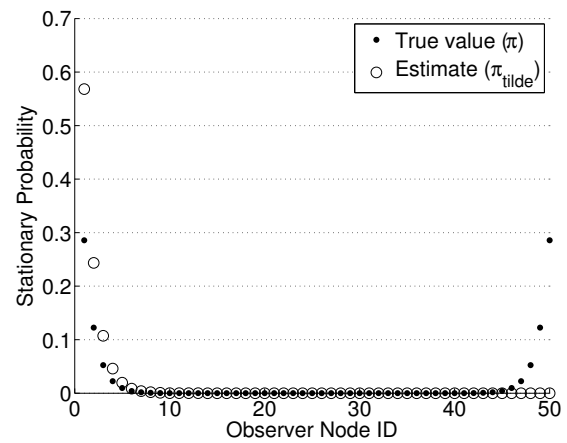
(c) MM1 Queue.



(d) MM1 Queue.

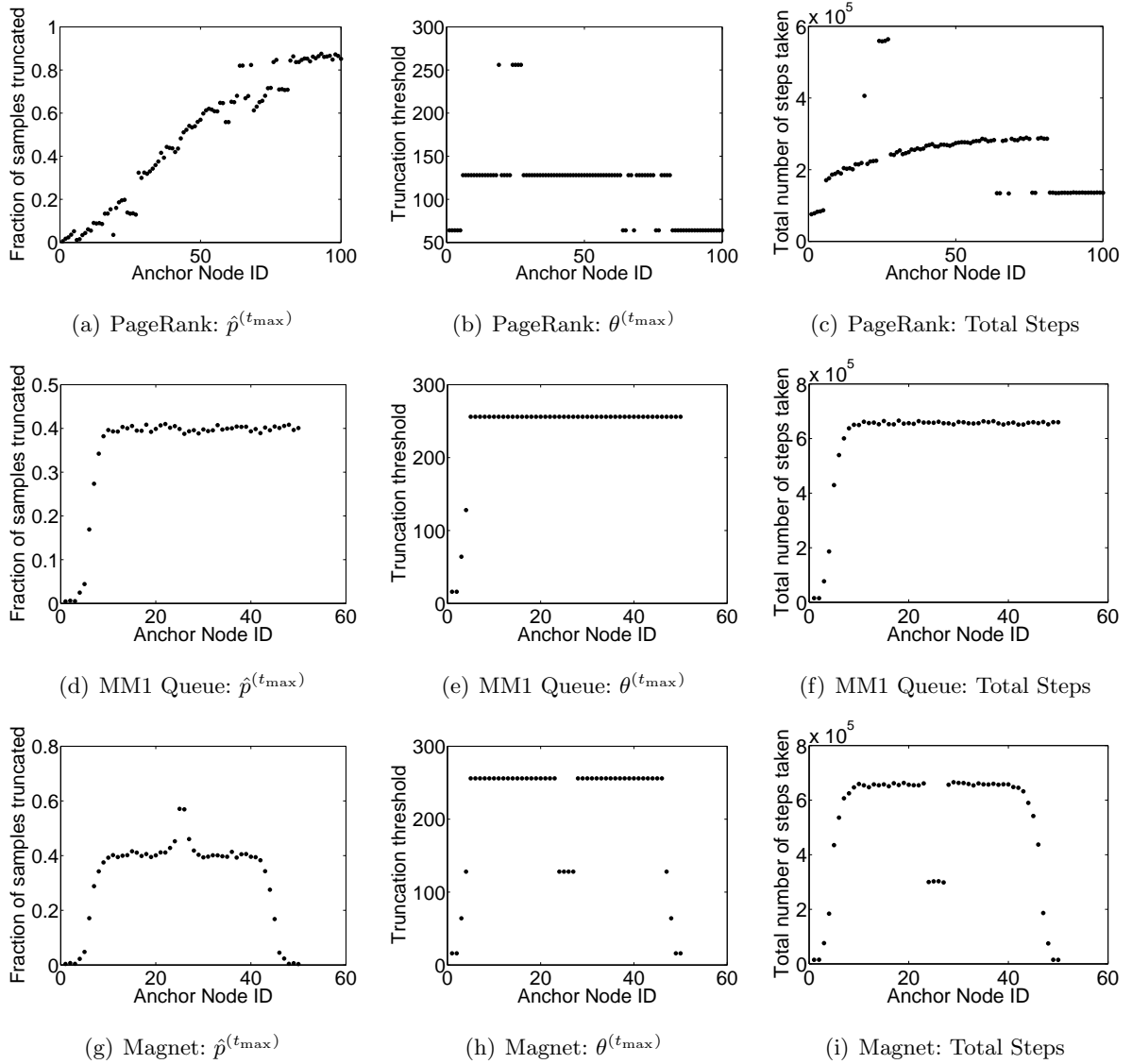


(e) Magnet.



(f) Magnet.

**Figure 3** These plots show the estimates that our algorithm obtains for three different Markov chains.



**Figure 4** This shows the value of three variables from the final iteration of the algorithm: fraction of samples truncated =  $\hat{p}^{(t_{\max})}$ ; truncation threshold =  $\theta^{(t_{\max})}$ ; and total number of random walk steps taken =  $N^{(t_{\max})} \cdot \hat{T}_i^{(t_{\max})}$ .

the fact that the random samples have close to zero probability of sampling from the opposite half of the graph. Therefore the estimates are computed without being able to observe the opposite half of the graph. As the challenge is due to the poor mixing properties of the graph, both  $\hat{\pi}_i$  and  $\tilde{\pi}_i$  are poor estimates. In the figure, it is difficult to distinguish the two estimates because they are nearly the same and thus superimposed upon each other. We compute the fundamental matrix  $Z$  for this Markov chain, and find that for most pairs  $i, j \in \Sigma$ ,  $|Z_{ij}|$  is on the order of  $10^9$ .

Standard methods such as power iteration or MCMC will also perform poorly on this graph, as it would take an incredibly large amount of time for the random walk to fully mix across the middle

border. The final outputs of both methods are very sensitive to the starting vector, since with high probability each random walk will stay on the half of the graph in which it was initialized. The estimates are neither guaranteed to be upper or lower bounds upon the true stationary probability.

Figure 4 plots the quantities  $\hat{p}^{(t_{\max})}$ ,  $\theta^{(t_{\max})}$ , and  $N^{(t_{\max})} \cdot \hat{T}_i^{(t_{\max})}$  for the execution of our algorithm on these three examples as a function of the chosen anchor node. We observe a correlation between these quantities and the stationary probability of a node. Consider the results for the MM1 queue Markov chain shown in Figures 4(d), 4(e), and 4(f). The fraction of samples truncated, the truncation threshold, and the total computation time increase as  $\pi_i$  decreases, and then flattens out for nodes  $i$  greater than 10. This illustrates the effect of the truncation and termination conditions. For nodes with small  $\pi_i$ , although  $\mathbb{E}_i[T_i]$  is large, because of the termination condition involving  $\Delta$ , the algorithm will terminate quickly, not wasting extra time to obtain unnecessary precision.

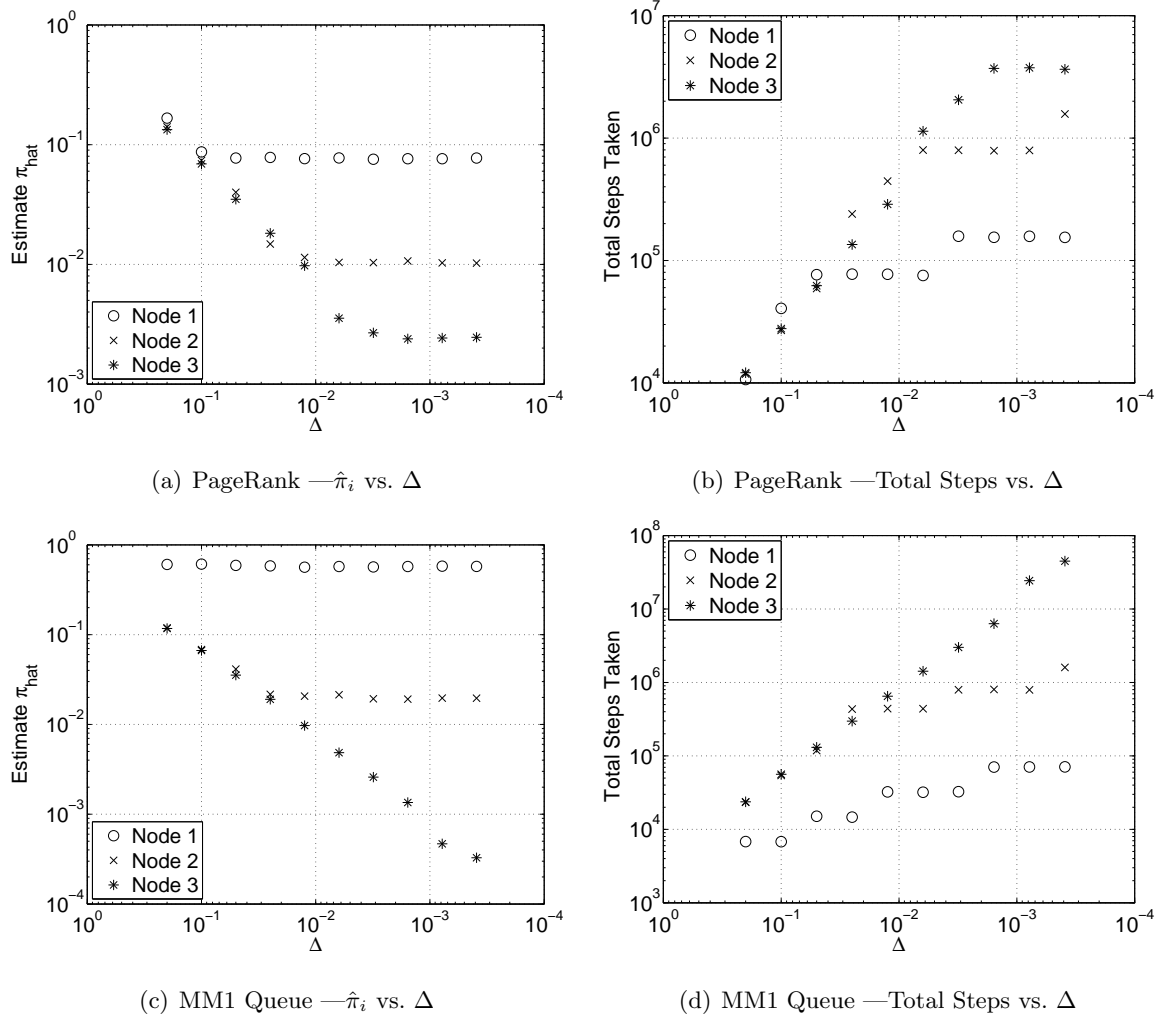
## 9.2. Algorithm Results for Observer Nodes

Figures 3(b), 3(d), and 3(f) plot results of a single execution of the algorithm when the anchor node is fixed to be node 1. The estimates  $\tilde{\pi}_j^{(t_{\max})}$  for a set of observer nodes is plotted along side the true value for  $\pi_j$ . In the PageRank and Magnet Markov chains,  $J = \Sigma$ , and for the MM1 queue Markov chain,  $J$  is the first 50 nodes.

Figures 3(b) and 3(d) both show that the estimates  $\tilde{\pi}_j$  are indeed close to  $\pi_j$ . As expected, Figure 3(f) shows that the algorithm overestimates the leftmost nodes by a factor of two, and completely does not detect the nodes on the right. Again, this is due to the poor mixing of the Markov chain such that the probability of visiting the opposite half of the graph is very small. The severity of this effect will depend on the choice of  $q_1$  and  $q_2$ . This is a graph on which any local algorithm will perform poorly on, since the algorithm will not be aware of the second attracting node on the opposite end.

## 9.3. Algorithm performance as a function of $\Delta$

Figure 5 shows the results of our algorithm as a function of  $\Delta$ . Recall that parameter  $\Delta$  only affects the termination conditions of the algorithm. The plot shows results from three separate executions of the algorithm with different anchor nodes, chosen to illustrate the behavior on nodes with varying  $\pi_i$ . Figures 5(a) and 5(c) plot the estimates for the stationary probability, and Figures 5(b) and 5(d) plot the total steps taken in the last iteration of the algorithm, which we recall is of the same order as the total random walk steps over all iterations. The figures are shown on a log-log scale. The results confirm that the computation time of the algorithm is upper bounded by  $O(1/\Delta)$ , which is linear when plotted on log-log scale. When  $\Delta > \pi_i$ , then  $\hat{\pi}_i$  is effectively lower bounded at  $\Delta$ , and the computation time behaves as  $1/\Delta$ . When  $\Delta < \pi_i$ , then  $\hat{\pi}_i$  concentrates



**Figure 5** These figures show the dependence of our algorithm on the parameter  $\Delta$ .

around  $\pi_i$  and no longer decreases with  $\Delta$ . The computation time levels off and grows slower than  $O(1/\Delta)$ .

## 10. Discussion: Understanding Properties of PageRank

Equation (12) shows that the PageRank random walk is effectively a convex combination between a directed graph given by  $A$ , and a complete graph. We can thus analyze the tradeoff as a function of  $\beta$ , a tuning parameter that controls whether the PageRank Markov chain is closer to the underlying directed graph or to the complete graph. The existing algorithms and analysis for computing PageRank locally (Jeh and Widom 2003, Fogaras et al. 2005, Avrachenkov et al. 2007, Bahmani et al. 2010, Borgs et al. 2012) only utilize the property that with probability  $\beta$ , the transition is chosen from the complete graph. When  $\beta$  is large, the Markov chain mixes quickly due to the complete graph, and these algorithms also performs efficiently; however, when  $\beta$  is small, regardless

of the properties of the underlying graph, the algorithm scales with  $1/\beta$ , even though the Markov chain might still mix quickly depending on the underlying graph.

For the following simulations, we restrict ourselves to the personalized PageRank setting. For a fixed node  $x$  and scalar  $\beta$ , we consider the Markov chain with transition probability matrix  $\tilde{P}$ :

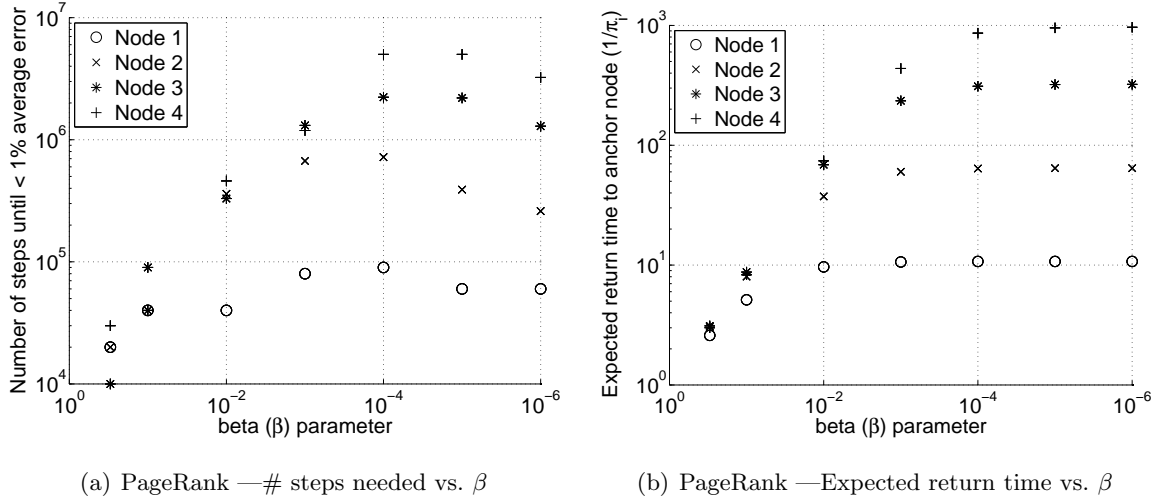
$$\tilde{P} = \beta \mathbf{1} \cdot e_x^T + (1 - \beta) D^{-1} A, \quad (13)$$

where  $A$  is the adjacency matrix of the underlying graph, generated according to the configuration model with a power law degree distribution, and  $D$  is a diagonal matrix of the out-degrees. In every step, there is a probability  $\beta$  of returning to the node  $x$ , and probability  $1 - \beta$  of choosing a neighbor at random (according to the configuration model generated graph). We choose four different nodes in the graph to be node  $x$ , and several different values for  $\beta$ . For each combination of  $x$  and  $\beta$ , we consider the number of random walk steps required to estimate  $\pi_x$  in the Markov chain corresponding to transition probability matrix  $\tilde{P}$ . This allows us to compare our algorithm with the existing PageRank algorithm presented by Fogaras et al. (2005) and Avrachenkov et al. (2007).

Fogaras' algorithm takes geometric length random walk samples on the underlying graph given by matrix  $A$ , beginning at node  $x$ . Our proposed algorithm samples random walks over  $\tilde{P}$  that begin and end at node  $x$ . Sample paths in each of these algorithms can be considered as adjacent contiguous sections of one long sample path over  $\tilde{P}$ , which contains the full information that each algorithm collects about the graph. We would like to minimize the total number of random walk steps taken (across all sample paths of various lengths).

In the following simulation, we sample ten long random walks starting at node  $x$  having five million steps each. After every ten thousand steps, we compute the current fraction of visits to node  $x$  out of the total steps taken thus far. This quantity converges to  $\pi_x$  as the length goes to infinity. The percentage error between the estimate and the true  $\pi_x$  is averaged over all ten samples. Then we find the smallest number of steps such that the average error is less than 1%. This is shown in Figure 6(a) as a function of  $\beta$  for four different chosen nodes  $x$ . The expected return time to node  $x$  in each of these settings is shown in Figure 6(b). Both figures are shown on log-log scale.

When  $\beta$  is large, the  $\beta$  jump back to  $x$  dominates the random walk, causing the random walk to behave similarly for different chosen nodes  $x$ . In this setting  $\beta$  can be used to determine the required number of samples to achieve good accuracy. When  $\beta$  is small, the random walk approaches that of the underlying graph, thus the choice of node  $x$  greatly affects the computation. The number of steps to achieve 1% accuracy does not increase much as  $\beta$  decreases for nodes  $x$  that have large  $\pi_x$  in the underlying graph (such as node 1). However, for nodes such that  $\pi_x$  is small in the underlying



**Figure 6** These figures show the number of random walk steps needed to accurately estimate the stationary probability of a node, as well as the expected return time to a node in a PageRank Markov chain, for four different nodes, and as a function of the parameter  $\beta$ .

graph (such as node 4), the required number of steps increases significantly as  $\beta$  decreases. We observe a relationship between  $\mathbb{E}_x[T_x]$  and the total amount of computation steps needed for 1% accuracy. This clearly highlights that in the setting when  $\beta$  is very small, it becomes critical to consider information from the underlying graph in determining the number of steps to sample. The existing algorithm only uses the parameter  $\beta$  to govern the number of steps taken for the algorithm, where the length of a sample random walk scales with  $1/\beta$ . However, for small  $\beta$ , this is unnecessarily expensive, as some nodes (such as node 1) do not in fact require the number of sample steps to increase with  $1/\beta$ . Our proposed algorithm relies on the expected return time to node  $x$  as the length of each sample path, which adjusts for nodes that have varying importances in the underlying graph.

## 11. Conclusion

This paper presents a local algorithm for estimating the stationary probabilities of a subset of nodes in a Markov chain. The algorithm is a *truncated Monte Carlo* method, sampling return paths to the anchor node. The key insight it utilizes is that the stationary probability of a node is inversely proportional to the expected return time to the node  $\mathbb{E}_i[T_i]$ . In order to obtain estimates for the observer nodes  $j \neq i$ , it uses the property that the stationary probability  $\pi_j$  is equal to the fraction of expected visits to node  $j$  along the return path to node  $i$ .

We provide theoretical guarantees for the tightness of approximation and computation time of our algorithm. We guarantee that the estimate  $\hat{\pi}_i^{(t_{\max})}$ , is an upper bound for  $\pi_i$  with high probability. For Markov chains that mix well, the estimate will be tight with high probability for nodes such



that  $\pi_i > \Delta$ . The analyses and guarantees rely on the property that for many positive recurrent countable state space Markov chains, the distribution over return times to a node concentrates around the mean. We showed that this concentration rate is related to the mixing properties of the Markov chain. The computation time of the algorithm is upper bounded by  $O(\ln(1/\alpha)/\epsilon^3\Delta)$  with high probability. Since the computation time is constant with respect to the Markov chain, this algorithm is suitable for Markov chains with large state spaces. The algorithm has many practical benefits. It can be implemented easily in a distributed and parallelized fashion, and it only uses a constant size neighborhood around the anchor node, upper bounded by  $1/\Delta$ . The computation only involves counting and taking an average, thus it is simple and memory efficient.

### Appendix A: Proofs of Preliminary Results

The proofs of Lemmas 3 to 6 use the following fact:

FACT 1. If  $\sum_{k=1}^t x_k \leq 1$ , then  $\prod_{k=1}^t (1 - x_k) \geq 1 - \sum_{k=1}^t x_k$ .

*Proof of Lemma 3.* Let  $A_t$  denote the event  $\{\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]\}$ . Since  $N^{(t)}$  is a random variable due to its dependence on  $\hat{T}_i^{(t-1)}$ , the distribution of  $\hat{T}_i^{(t)}$  depends on the value of  $\hat{T}_i^{(t-1)}$ . Conditioned on the event  $A_{t-1}$ ,

$$\begin{aligned} N^{(t)} &= \left\lceil \frac{3(1+\epsilon)\theta^{(t)} \ln(4\theta^{(t)}/\alpha)}{\hat{T}_i^{(t-1)} \epsilon^2} \right\rceil \\ &\geq \frac{3(1+\epsilon)\theta^{(t)} \ln(4\theta^{(t)}/\alpha)}{(1+\epsilon)\mathbb{E}_i[\hat{T}_i^{(t-1)}] \epsilon^2} = \frac{3\theta^{(t)} \ln(4\theta^{(t)}/\alpha)}{\mathbb{E}_i[\hat{T}_i^{(t-1)}] \epsilon^2}. \end{aligned} \quad (14)$$

Then we apply Chernoff's bound for independent identically distributed bounded random variables (cf. Theorem 11), substitute in for  $N^{(t)}$ , and use the facts that  $\mathbb{E}_i[\hat{T}_i^{(t)}] \geq \mathbb{E}_i[\hat{T}_i^{(t-1)}]$  and  $\theta^{(t)} = 2^t$  for all  $t$ , to show that

$$\begin{aligned} \mathbb{P}_i(\neg A_t | A_{t-1}) &\leq 2 \exp\left(-\frac{\epsilon^2 N^{(t)} \mathbb{E}_i[\hat{T}_i^{(t)}]}{3\theta^{(t)}}\right) \\ &\leq 2 \exp\left(-\frac{\mathbb{E}_i[\hat{T}_i^{(t)}] \ln(4\theta^{(t)}/\alpha)}{\mathbb{E}_i[\hat{T}_i^{(t-1)}]}\right) \\ &\leq \frac{\alpha}{2\theta^{(t)}} \leq \frac{\alpha}{2^{t+1}}. \end{aligned} \quad (15)$$

It can be verified that  $\mathbb{P}_i(\neg A_1)$  is similarly upper bounded by  $\frac{\alpha}{4}$  using the definition of  $N^{(1)}$ . Therefore, by Bayes rule and by the fact that  $\hat{T}_i^{(t)}$  is independent from  $\hat{T}_i^{(k)}$  conditioned on  $\hat{T}_i^{(t-1)}$  for all  $k < t$ , we show that

$$\mathbb{P}_i\left(\bigcap_{k=1}^t A_k\right) = \mathbb{P}_i(A_1) \prod_{k=2}^t \mathbb{P}_i(A_k | A_{t-1}) \geq \prod_{k=1}^t \left(1 - \frac{\alpha}{2^{k+1}}\right). \quad (16)$$

By applying Fact 1, it follows that

$$\mathbb{P}_i \left( \bigcap_{k=1}^t A_k \right) \geq 1 - \sum_{k=1}^t \frac{\alpha}{2^{k+1}} = 1 - \frac{\alpha}{2} (1 - 2^{-t}) \geq 1 - \alpha. \quad (17)$$

□

*Proof of Lemma 4.* Let  $A_t$  denote the event  $\{\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]\}$ . Let  $B_t$  denote the event  $\{\hat{p}^{(t)} \in \mathbb{P}_i(T_i > \theta^{(t)}) \pm \frac{\epsilon}{3}\}$ .  $\hat{T}_i^{(t)}$  is independent from  $\hat{p}_i^{(t-1)}$  conditioned on  $\hat{T}_i^{(t-1)}$ . Therefore, it follows from (15) that

$$\mathbb{P}_i(\neg A_t | A_{t-1} \cap B_{t-1}) \leq \frac{\alpha}{2^{t+1}}.$$

By applying Hoeffding's Inequality for Bernoulli random variables to  $\hat{p}^{(k)}$  and substituting (14), we show that

$$\begin{aligned} \mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) &\leq 2 \exp\left(-\frac{2}{9}\epsilon^2 N^{(t)}\right) \\ &\leq 2 \exp\left(-\frac{2\theta^{(t)} \ln(4\theta^{(t)}/\alpha)}{3\mathbb{E}_i[\hat{T}_i^{(t-1)}]}\right). \end{aligned}$$

Since  $\mathbb{E}_i[\hat{T}_i^{(t-1)}] \leq \theta^{(t-1)} = \theta^{(t)}/2$ ,

$$\mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) \leq 2 \exp\left(-\frac{4}{3} \ln(4\theta^{(t)}/\alpha)\right) \leq \frac{\alpha}{2\theta^{(t)}} \leq \frac{\alpha}{2^{t+1}}.$$

By applying the union bound, it follows that

$$\mathbb{P}_i(A_t \cap B_t | A_{t-1} \cap B_{t-1}) \geq 1 - \frac{\alpha}{2^t}. \quad (18)$$

We can easily verify using the same techniques that

$$\mathbb{P}_i(A_1 \cap B_1) \geq 1 - \frac{\alpha}{2}.$$

Therefore,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t (B_k \cap A_k) \right) \geq \prod_{k=1}^t \left(1 - \frac{\alpha}{2^k}\right).$$

We use Fact 1 to show that

$$\mathbb{P}_i \left( \bigcap_{k=1}^t (B_k \cap A_k) \right) \geq 1 - \sum_{k=1}^t \frac{\alpha}{2^k} = 1 - \alpha(1 - 2^{-t}) \geq 1 - \alpha.$$

□

*Proof of Lemma 5.* Recall that  $t_0$  is defined such that  $\mathbb{P}_i(T_i > \theta^{(t_0)}) < \frac{1}{2}$ . Let  $A_t$  denote the event

$\{\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]\}$ . Let  $B_t$  denote the event  $\{(1 - \hat{p}^{(t)}) \in (1 \pm \epsilon)(1 - \mathbb{P}(T_i > \theta^{(t)}))\}$ .

$$\mathbb{P}_i \left( \bigcap_{k=t_0}^t B_k \bigcap_{k=1}^t A_k \right) = \mathbb{P}_i \left( \bigcap_{k=t_0}^t (B_k \cap A_k) \middle| \bigcap_{k=1}^{t_0-1} A_k \right) \mathbb{P}_i \left( \bigcap_{k=1}^{t_0-1} A_k \right). \quad (19)$$

$\hat{T}_i^{(t)}$  and  $\hat{p}^{(t)}$  are not independent from  $\hat{T}_i^{(t-1)}$  and  $\hat{p}^{(t-1)}$  due to the dependence of  $N^{(t)}$  as a random variable upon  $\hat{T}_i^{(t-1)}$ . However,  $\hat{T}_i^{(t)}$  is independent from  $\hat{p}_i^{(t-1)}$  conditioned on  $\hat{T}_i^{(t-1)}$ . Therefore, it follows from (15) that

$$\mathbb{P}_i(\neg A_t | A_{t-1} \cap B_{t-1}) \leq \frac{\alpha}{2^{t+1}}. \quad (20)$$

By using Chernoff's bound for Bernoulli random variables and substituting (14), we show that

$$\begin{aligned} \mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) &\leq 2 \exp\left(-\frac{\epsilon^2 N^{(t)}(1 - \mathbb{P}_i(T_i > \theta^{(t)}))}{3}\right) \\ &\leq 2 \exp\left(-\frac{\theta^{(t)} \ln(4\theta^{(t)}/\alpha)(1 - \mathbb{P}_i(T_i > \theta^{(t)}))}{\mathbb{E}_i[\hat{T}_i^{(t-1)}]}\right). \end{aligned} \quad (21)$$

By definition,  $\mathbb{E}_i[\hat{T}_i^{(t-1)}] \leq \theta^{(t-1)} = \theta^{(t)}/2$ . Since we are given that  $\mathbb{P}_i(T_i > \theta^{(t)}) \leq \frac{1}{2}$ , then

$$\frac{\theta^{(t)}(1 - \mathbb{P}_i(T_i > \theta^{(t)}))}{\mathbb{E}_i[\hat{T}_i^{(t-1)}]} \geq 1. \quad (22)$$

By substituting (22) into (21), it follows that

$$\mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) \leq \frac{2\alpha}{4\theta^{(t)}} \leq \frac{\alpha}{2^{t+1}}. \quad (23)$$

We combine (20) and (23), and apply the union bound to show that

$$\mathbb{P}_i(A_t \cap B_t | A_{t-1} \cap B_{t-1}) \geq 1 - \frac{\alpha}{2^t}. \quad (24)$$

We can easily verify using the same techniques that

$$\mathbb{P}_i\left(A_{t_0} \cap B_{t_0} \left| \bigcap_{k=1}^{t_0-1} A_k\right.\right) \geq 1 - \frac{\alpha}{2^{t_0}}.$$

We use Bayes' rule and substitute (16) and (24) into (19) to show that

$$\mathbb{P}_i\left(\bigcap_{k=t_0}^t B_k \bigcap_{k=1}^t A_k\right) \geq \prod_{k=t_0}^t \left(1 - \frac{\alpha}{2^k}\right) \prod_{k=1}^{t_0-1} \left(1 - \frac{\alpha}{2^{k+1}}\right).$$

We apply Fact 1 to show that

$$\mathbb{P}_i\left(\bigcap_{k=t_0}^t B_k \bigcap_{k=1}^t A_k\right) \geq 1 - \sum_{k=1}^t \frac{\alpha}{2^k} + \sum_{k=1}^{t_0} \frac{\alpha}{2^{k+1}} = 1 - \alpha(1 - 2^{-t}) + \sum_{k=1}^{t_0} \frac{\alpha}{2^{k+1}} \geq 1 - \alpha.$$

□

*Proof of Lemma 6.* Let  $A_t$  denote the event  $\{\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]\}$ . Let  $B_t$  denote the event  $\{\hat{F}_j^{(k)} \in \mathbb{E}_i[\hat{F}_j^{(k)}] \pm \epsilon \mathbb{E}_i[\hat{T}_i^{(k)}]\}$ .  $\hat{T}_i^{(t)}$  is independent from  $\hat{F}_j^{(t-1)}$  conditioned on  $\hat{T}_i^{(t-1)}$ . Therefore, it follows from (15) that

$$\mathbb{P}_i(\neg A_t | A_{t-1} \cap B_{t-1}) \leq \frac{\alpha}{2^{t+1}}.$$

Recall from the definition of  $\hat{F}_j^{(t)}$  that  $\hat{F}_j^{(t)} \in [0, \theta^{(t)}]$ . Therefore, by applying Chernoff's bound for independent identically distributed bounded random variables, and substituting (14), we show that

$$\begin{aligned} \mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) &\leq 2 \exp \left( - \left( \frac{\epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[\hat{F}_j^{(t)}]} \right)^2 \frac{N^{(t)} \mathbb{E}_i[\hat{F}_j^{(t)}]}{3\theta^{(t)}} \right) \\ &\leq 2 \exp \left( - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]^2 \ln(4\theta^{(t)}/\alpha)}{\mathbb{E}_i[\hat{F}_j^{(t)}] \mathbb{E}_i[\hat{T}_i^{(t-1)}]} \right). \end{aligned} \quad (25)$$

Since  $\mathbb{E}_i[\hat{T}_i^{(t)}] \geq \mathbb{E}_i[\hat{T}_i^{(t-1)}]$  and  $\mathbb{E}_i[\hat{T}_i^{(t)}] \geq \mathbb{E}_i[\hat{F}_j^{(t)}]$ , then

$$\frac{\mathbb{E}_i[\hat{T}_i^{(t)}]^2}{\mathbb{E}_i[\hat{F}_j^{(t)}] \mathbb{E}_i[\hat{T}_i^{(t-1)}]} \geq 1.$$

By substituting into (25), it follows that

$$\mathbb{P}_i(\neg B_t | A_{t-1} \cap B_{t-1}) \leq 2 \frac{\alpha}{4\theta^{(t)}} \leq \frac{\alpha}{2^{t+1}}.$$

By applying the union bound, we show that

$$\mathbb{P}_i(A_t \cap B_t | A_{t-1} \cap B_{t-1}) \geq 1 - \frac{\alpha}{2^t}. \quad (26)$$

We can easily verify using the same techniques that

$$\mathbb{P}_i(A_1 \cap B_1) \geq 1 - \frac{\alpha}{2}.$$

Therefore,

$$\mathbb{P}_i \left( \bigcap_{k=1}^t (B_k \cap A_k) \right) \geq \prod_{k=1}^t \left( 1 - \frac{\alpha}{2^k} \right).$$

We use the Fact 1 to show that

$$\mathbb{P}_i \left( \bigcap_{k=1}^t (B_k \cap A_k) \right) \geq 1 - \sum_{k=1}^t \frac{\alpha}{2^k} = 1 - \alpha(1 - 2^{-t}) \geq 1 - \alpha.$$

□

*Proof of Lemma 7.* This proof is adapted from a proof found in Aldous and Fill. It is included here for completeness.

$$\begin{aligned} \mathbb{P}_i(T_i > m\alpha) &= \prod_{s=0}^{m-1} \mathbb{P}_i(T_i > (s+1)\alpha \mid T_i > s\alpha) \\ &= \prod_{s=0}^{m-1} \left( \sum_{j \in \Sigma} \mathbb{P}_i(T_i > (s+1)\alpha \mid X_{s\alpha} = j, T_i > s\alpha) \mathbb{P}_i(X_{s\alpha} = j \mid T_i > s\alpha) \right). \end{aligned}$$

By the Markov property,

$$\mathbb{P}_i(T_i > m\alpha) = \prod_{s=0}^{m-1} \left( \sum_{j \in \Sigma} \mathbb{P}_j(T_i > \alpha) \mathbb{P}_i(X_{s\alpha} = j \mid T_i > s\alpha) \right).$$

By Markov's inequality,

$$\begin{aligned} \mathbb{P}_i(T_i > m\alpha) &\leq \prod_{s=0}^{m-1} \left( \sum_{j \in \Sigma} \left( \frac{\mathbb{E}_j[T_i]}{\alpha} \right) \mathbb{P}_i(X_{s\alpha} = j \mid T_i > s\alpha) \right) \\ &\leq \prod_{s=0}^{m-1} \left( \frac{\max_{j \in \Sigma} \mathbb{E}_j[T_i]}{\alpha} \right) = \left( \frac{H_i}{\alpha} \right)^m. \end{aligned}$$

By choosing  $\alpha = 2H_i$  and  $m = \lfloor k/2H_i \rfloor$ , it follows that

$$\mathbb{P}_i(T_i > k) \leq \mathbb{P}_i \left( T_i > \left\lfloor \frac{k}{2H_i} \right\rfloor \cdot 2H_i \right) \leq 2^{-\lfloor k/2H_i \rfloor} \leq 2 \cdot 2^{-k/2H_i}.$$

□

## Appendix B: Proofs for Estimation Error

*Proof of Theorem 2.* By Lemma 3, with probability greater than  $1 - \alpha$ ,  $\hat{T}_i^{(t)} \in (1 \pm \epsilon) \mathbb{E}_i[\hat{T}_i^{(t)}]$  for all  $t$ . Therefore it follows that with probability greater than  $1 - \alpha$ ,

$$\begin{aligned} \left| 1 - \frac{\pi_i}{\hat{\pi}_i^{(t)}} \right| &\leq \max \left( 1 - (1 - \epsilon) \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]}, (1 + \epsilon) \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} - 1 \right) \\ &= \frac{\epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} + \max \left( 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]}, \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} - 1 \right). \end{aligned}$$

Since  $\mathbb{E}_i[\hat{T}_i^{(t)}] \leq \mathbb{E}_i[T_i]$ ,

$$\left| 1 - \frac{\pi_i}{\hat{\pi}_i^{(t)}} \right| \leq \frac{\epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} + 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} = (1 - \epsilon) \left( 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} \right) + \epsilon \quad (27)$$

By multiplying (6) by  $\mathbb{E}[\hat{T}_i^{(t)}]$ , and by loosely upper bounding  $(Z_{ii} - Z_{qi})$  by  $Z_{\max}(i)$  for all  $q$ , we obtain

$$1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} \leq 2Z_{\max}(i) \mathbb{P}_i \left( T_i^{(t)} > \theta^{(t)} \right). \quad (28)$$

Substitute (28) into (27), and apply Lemma 7 to complete the first statement of the proof. Next, we need to show that when the algorithm terminates at condition (b),  $\mathbb{P}(T_i > \theta) \leq \epsilon(\frac{4}{3} + \epsilon)$  with high probability. When the algorithm terminates at condition (b),  $\hat{p}^{(t)} \hat{\pi}_i^{(t)} < \epsilon \Delta$  and  $\hat{\pi}_i^{(t)} \geq \frac{\Delta}{1 + \epsilon}$ . Therefore,  $\hat{p}^{(t)} < \epsilon(1 + \epsilon)$ . By Lemma 4, with probability greater than  $1 - \alpha$ ,  $\mathbb{P}_i(T_i > \theta^{(t)}) \leq \hat{p}^{(t)} + \frac{\epsilon}{3} \leq \epsilon(\frac{4}{3} + \epsilon)$ . Therefore, it follows that

$$\left| \frac{\hat{\pi}_i^{(t)} - \pi_i}{\hat{\pi}_i^{(t)}} \right| \leq 2\epsilon(1 - \epsilon) \left( \frac{4}{3} + \epsilon \right) Z_{\max}(i) + \epsilon \leq \epsilon(3Z_{\max}(i) + 1).$$

□

*Proof of Theorem 3.* By Lemma 5, with probability greater than  $1 - \alpha$ ,  $\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]$  for all  $t$ , and  $(1 - \hat{p}^{(t)}) \in (1 \pm \epsilon)(1 - \mathbb{P}_i(T_i > \theta^{(t)}))$  for all  $t$  such that  $\mathbb{P}_i(T_i > \theta^{(t)}) > \frac{1}{2}$ . Therefore, with probability greater than  $1 - \alpha$ , for every  $t$  such that  $\mathbb{P}_i(T_i > \theta^{(t)}) > \frac{1}{2}$ ,

$$\begin{aligned} & \left| 1 - \frac{\hat{T}_i^{(t)}}{(1 - \hat{p}^{(t)})\mathbb{E}_i[T_i]} \right| \\ & \leq \max \left( 1 - \frac{(1 - \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 + \epsilon)(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]}, \frac{(1 + \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \epsilon)(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} - 1 \right) \\ & \leq \max \left( \frac{1 - \epsilon}{1 + \epsilon} \left( 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} \right), \frac{1 + \epsilon}{1 - \epsilon} \left( \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} - 1 \right) \right) + \frac{2\epsilon}{1 - \epsilon} \\ & \leq \frac{1 + \epsilon}{1 - \epsilon} \left| 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} \right| + \frac{2\epsilon}{1 - \epsilon}. \end{aligned} \quad (29)$$

By rearranging (10), and upper bounding  $(Z_{ii} - Z_{qi})$  by  $Z_{\max}(i)$  for all  $q$ , it follows that

$$\left| 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} \right| \leq \max(2Z_{\max}(i) - 1, 1) \left( \frac{\mathbb{P}_i(T_i^{(t)} > \theta^{(t)})}{1 - \mathbb{P}_i(T_i^{(t)} > \theta^{(t)})} \right). \quad (30)$$

Substitute (30) into (29), and apply Lemma 7 to complete the proof.  $\square$

In order to analyze the distribution over the number of visits to node  $j$  on a return path to node  $i$ , we will use the following Lemma as stated by Aldous and Fill (1999).

**LEMMA 14 (Aldous, Fill).** *For distinct  $i, k \in \Sigma$ , the expected number of visits to a node  $j$  beginning from node  $k$  before visiting  $i$  is equal to*

$$\mathbb{E}_k \left[ \sum_{t=1}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \right] = \pi_j (E_k[T_i] + E_i[T_j] - E_k[T_j]).$$

*Proof of Lemma 11.* By definition,

$$\begin{aligned} & \mathbb{E}_i[F_j] - \mathbb{E}_i[\hat{F}_j^{(t)}] \\ & = \mathbb{E}_i \left[ \sum_{t=1}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \right] - \mathbb{E}_i \left[ \sum_{t=1}^{\theta^{(t)}} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \right] \\ & = \mathbb{P}(T_i > \theta^{(t)}) \mathbb{E}_i \left[ \sum_{t=\theta^{(t)}}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \middle| T_i > \theta^{(t)} \right] \\ & = \mathbb{P}(T_i > \theta^{(t)}) \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \mathbb{E}_i \left[ \sum_{t=\theta^{(t)}+1}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \middle| X_{\theta^{(t)}} = q, T_i > \theta^{(t)} \right] \\ & = \mathbb{P}(T_i > \theta^{(t)}) \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \mathbb{E}_q \left[ \sum_{t=1}^{\infty} \mathbb{1}\{X_t = j\} \mathbb{1}\{t \leq T_i\} \right]. \end{aligned}$$

We divide by  $\mathbb{E}_i[F_j]$  and use Lemma 14 and Lemma 1(a) to show that

$$\begin{aligned} & 1 - \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[F_j]} \\ &= \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[F_j]} \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \pi_j(E_q[T_i] + E_i[T_j] - E_q[T_j]) \\ &= \mathbb{P}(T_i > \theta^{(t)}) \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \frac{(E_q[T_i] + E_i[T_j] - E_q[T_j])}{\mathbb{E}_i[T_i]}. \end{aligned} \quad (31)$$

By multiplying (9) by  $\mathbb{E}_i[\hat{T}_i^{(t)}]$ , it follows that

$$1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} = \mathbb{P}_i(T_i > \theta^{(t)}) \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \frac{\mathbb{E}_q[T_i]}{\mathbb{E}_i[T_i]}. \quad (32)$$

We use (31) and (32) to show that

$$\begin{aligned} \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_j &= \frac{\mathbb{E}_i[F_j]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \left( \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[F_j]} - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} \right) \\ &= \frac{\mathbb{E}_i[F_j]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \mathbb{P}(T_i > \theta^{(t)}) \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \frac{(E_q[T_j] - E_i[T_j])}{\mathbb{E}_i[T_i]} \\ &= \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) \pi_j(E_q[T_j] - E_i[T_j]). \end{aligned}$$

By Lemma 2,

$$\frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_j = \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \left( \sum_{q \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = q | T_i > \theta^{(t)}) (Z_{ij} - Z_{qj}) \right).$$

□

*Proof of Theorem 4.* By Lemma 6, with probability greater than  $1 - \alpha$ ,  $\hat{F}_j^{(t)} \in \mathbb{E}_i[\hat{F}_j^{(t)}] \pm \epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]$  and  $\hat{T}_i^{(t)} \in (1 \pm \epsilon) \mathbb{E}_i[\hat{T}_i^{(t)}]$  for all  $t$ . Therefore with probability greater than  $1 - \alpha$ ,

$$\frac{(1 - \epsilon) \hat{F}_j^{(t)}}{\hat{T}_i^{(t)}} \leq \frac{\mathbb{E}_i[\hat{F}_j^{(t)}] + \epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]}.$$

Therefore,

$$\hat{\pi}_j^{(t)} \leq \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} + \epsilon + \epsilon \hat{\pi}_j^{(t)}.$$

Similarly,

$$\begin{aligned} \frac{(1 + \epsilon) \hat{F}_j^{(t)}}{\hat{T}_i^{(t)}} &\geq \frac{\mathbb{E}_i[\hat{F}_j^{(t)}] - \epsilon \mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \\ \hat{\pi}_j^{(t)} &\geq \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \epsilon - \epsilon \hat{\pi}_j^{(t)}. \end{aligned}$$

Therefore, with probability greater than  $1 - \alpha$ ,

$$\begin{aligned} \left| \tilde{\pi}_j^{(t)} - \pi_j \right| &\leq \max \left( \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} + \epsilon(1 + \hat{\pi}_j^{(t)}) - \pi_j, \pi_j - \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} + \epsilon(1 + \hat{\pi}_j^{(t)}) \right) \\ &= \left| \frac{\mathbb{E}_i[\hat{F}_j^{(t)}]}{\mathbb{E}_i[\hat{T}_i^{(t)}]} - \pi_j \right| + \epsilon(1 + \tilde{\pi}_j^{(t)}). \end{aligned}$$

Use Lemma 11 to show that with probability greater than  $1 - \alpha$ ,

$$\begin{aligned} \left| \tilde{\pi}_j^{(t)} - \pi_j \right| &\leq \left| \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \left( \sum_{k \in \Sigma \setminus \{i\}} \mathbb{P}_i(X_{\theta^{(t)}} = k | T_i > \theta^{(t)}) Z_{ij} - Z_{kj} \right) \right| + \epsilon(1 + \tilde{\pi}_j^{(t)}) \\ &\leq \frac{\mathbb{P}(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} 2Z_{\max}(j) + \epsilon(1 + \tilde{\pi}_j^{(t)}). \end{aligned}$$

Since  $\hat{T}_i^{(t)} \in (1 \pm \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]$ , it follows that

$$\left| \tilde{\pi}_j^{(t)} - \pi_j \right| \leq (1 + \epsilon)\mathbb{P}(T_i > \theta^{(t)}) 2Z_{\max}(j)\hat{\pi}_i^{(t)} + \epsilon(1 + \tilde{\pi}_j^{(t)}).$$

Apply Lemma 7 to complete the proof.  $\square$

### Appendix C: Proofs for Computation Time

*Proof of Theorem 6(a).* Let  $t$  be large enough such that

$$\theta^{(t)} = 2^t \geq \frac{2H_i}{\ln 2} \ln \left( \left( \frac{2}{1 - 2^{-1/2H_i}} \right) \left( \frac{1}{\pi_i} - \frac{(1 + \epsilon)}{(1 - \epsilon)\Delta} \right)^{-1} \right). \quad (33)$$

Since we are given that  $\pi_i < \frac{(1 - \epsilon)\Delta}{(1 + \epsilon)}$ , this expression is well defined. We apply Lemma 7 to Lemma 8 to show

$$\mathbb{E}_i[T_i] - \mathbb{E}_i[\hat{T}_i^{(t)}] \leq \sum_{k=\theta^{(t)}}^{\infty} 2 \cdot 2^{-k/2H_i} = \frac{2 \cdot 2^{-\theta^{(t)}/2H_i}}{1 - 2^{-1/2H_i}}. \quad (34)$$

By substituting (33) into (34), it follows that

$$\mathbb{E}_i[\hat{T}_i^{(t)}] \geq \mathbb{E}_i[T_i] - \left( \frac{1}{\pi_i} - \frac{(1 + \epsilon)}{(1 - \epsilon)\Delta} \right) = \frac{(1 + \epsilon)}{(1 - \epsilon)\Delta}. \quad (35)$$

By combining Lemma 3 and (33), with probability greater than  $1 - \alpha$ ,

$$\begin{aligned} \hat{\pi}_i^{(t)} &= \frac{1}{\hat{T}_i^{(t)}} \leq \frac{1}{(1 - \epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]}, \\ &\leq \frac{\Delta}{(1 + \epsilon)}, \end{aligned}$$

satisfying termination condition (a). Therefore, with probability greater than  $1 - \alpha$ , the number of iterations in the algorithm is bounded above by

$$t_{\max} \leq \ln \left( \frac{2H_i}{\ln 2} \ln \left( \left( \frac{2}{1 - 2^{-1/2H_i}} \right) \left( \frac{1}{\pi_i} - \frac{(1 + \epsilon)}{(1 - \epsilon)\Delta} \right)^{-1} \right) \right).$$

Combine this with Lemma 12 to finish the proof.  $\square$



*Proof of Theorem 6(b).* Let  $t$  be large enough such that

$$\theta^{(t)} = 2^t \geq \frac{2H_i}{\ln 2} \ln \left( 2\pi_i \left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right) \right). \quad (36)$$

By substituting (36) into Lemma 7, it follows that

$$\mathbb{P}_i(T_i > \theta^{(t)}) \leq \mathbb{E}_i[T_i] \left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right)^{-1}. \quad (37)$$

Substitute (36) into (34) to show that

$$\mathbb{E}_i[\hat{T}_i^{(t)}] \geq \mathbb{E}_i[T_i] - \mathbb{E}_i[T_i] \left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right)^{-1} (1-2^{-1/H_i})^{-1}. \quad (38)$$

We combine (37) and (38) to show that

$$\frac{\mathbb{P}_i(T_i > \theta^{(t)})}{\mathbb{E}_i[\hat{T}_i^{(t)}]} \leq \frac{1}{\left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right)^{-1} - (1-2^{-1/2H_i})^{-1}} = \frac{(1-\epsilon)\epsilon\Delta}{2}. \quad (39)$$

By applying (17) to (39), it follows that with probability greater than  $1 - \frac{\alpha}{2}$ ,

$$\mathbb{P}_i(T_i > \theta^{(t)}) \hat{\pi}_i^{(t)} = \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{\hat{T}_i^{(t)}} \leq \frac{\mathbb{P}_i(T_i > \theta^{(t)})}{(1-\epsilon)\mathbb{E}_i[\hat{T}_i^{(t)}]} \leq \frac{\epsilon\Delta}{2}.$$

By Markov's inequality, for each time  $t$ ,

$$\mathbb{P}(\hat{p}^{(t)} \leq 2\mathbb{P}_i(T_i > \theta^{(t)})) \geq \frac{1}{2}.$$

Therefore, with probability greater than  $\frac{1}{2}$ ,

$$\hat{p}^{(t)} \hat{\pi}_i^{(t)} \leq 2\mathbb{P}_i(T_i > \theta^{(t)}) \hat{\pi}_i^{(t)} \leq \epsilon\Delta,$$

and the algorithm will satisfy termination condition (b). If the algorithm does not terminate in this iteration, then the algorithm will terminate in the next iteration with probability greater than  $\frac{1}{2}$ . Thus after another additional  $\log_2(\frac{2}{\alpha})$  iterations, we can guarantee that with probability greater than  $1 - \frac{\alpha}{2}$ , the algorithm will terminate. We apply the union bound to show that since  $\theta^{(t)} = 2^t$ , with probability greater than  $1 - \alpha$ , the number of iterations needed in the algorithm is upper bounded by

$$\begin{aligned} t &\leq \log_2(2/\alpha) + \log_2 \left( \frac{2H_i}{\ln 2} \ln \left( 2\pi_i \left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right) \right) \right) \\ &= \log_2 \left( \frac{4H_i}{\alpha \ln 2} \ln \left( 2\pi_i \left( \frac{2}{(1-\epsilon)\epsilon\Delta} + \frac{1}{1-2^{-1/2H_i}} \right) \right) \right) \end{aligned}$$

□

## Appendix D: Proofs for Countable State Space Markov Chains

To prove that the return times concentrate for countably infinite state space Markov chains, we use Lyapunov function analysis introduced by Foster (1953). We establish that indeed return times have exponentially decaying tail even for countable-state space Markov chain as long as they satisfy Assumption 1.

**Useful notation.** We introduce formal notation for observing  $\{X_t\}_{t \geq 0}$  over the subset  $B$ . Let  $\{Y_t\}_{t \geq 0}$  be a Markov chain with state space  $B$ .  $\{Y_t\}$  is a subsequence of  $\{X_t\}$  constructed in the following way. Define the subsequence  $\{S_k\}$  of  $\mathbb{Z}_+$  as:

$$S_0 \triangleq 0, S_k \triangleq \min\{t > S_{k-1} : X_t \in B\},$$

and define  $Y_t \triangleq X_{S_t}$  such that

$$P_Y(x, y) = \mathbb{P}(X_{S_1} = y | X_0 = x) \quad \text{for any } x, y \in B.$$

Let  $Q_t \triangleq S_{t+1} - S_t$ , the length of the path between  $X_{S_t}$  and  $X_{S_{t+1}}$ .

Let  $T_i^B \triangleq \inf\{t \geq 1 \mid Y_t = i\}$ , the return time to  $i$  for the chain  $\{Y_t\}$ .

Let  $H_i^B \triangleq \max_{j \in B} \mathbb{E}_j [T_i^B]$ , the maximal expected hitting time to node  $i$  for the chain  $\{Y_t\}$ . We can use these variables to express the return time to node  $i$  by

$$T_i = S_{T_i^B} = \sum_{k=0}^{T_i^B-1} Q_k. \quad (40)$$

LEMMA 15. *Let  $\{X_t\}$  be a countable state space Markov chain satisfying Assumption 1. Let  $\{Y_t\}$  be defined above as the Markov chain restricted to  $B$ , and let  $Q_k$  be the length between visits to  $B$ . For any  $W \in \mathbb{Z}_+$ ,  $Z \geq W$ , and  $i \in B$ ,*

$$\mathbb{P}_i \left( \sum_{k=0}^{W-1} Q_k > Z \right) \leq \exp \left( \frac{0.8(1-\rho)}{e^{\eta\nu_{\max}}} \left( \frac{1.25W e^{2\eta\nu_{\max}}}{(1-\rho)(e^{\eta\nu_{\max}} - \rho)} + W - Z \right) \right).$$

The constants  $\gamma$  and  $\nu_{\max}$  are given by the Assumption 1, and the scalars  $\eta$  and  $\rho$  are functions of  $\gamma$  and  $\nu_{\max}$ , as defined in (44) in Appendix F.

*Proof of Lemma 15.* By the law of iterated expectation,

$$\mathbb{E}_i \left[ \prod_{k=0}^{W-1} \exp(\lambda Q_k) \right] = \mathbb{E}_i \left[ \exp(\lambda Q_0) \mathbb{E}_i \left[ \prod_{k=1}^{W-1} \exp(\lambda Q_k) \middle| Q_0 \right] \right]$$

Conditioned on  $Y_1 = j$ ,  $Y_0$  and  $Q_0$  are independent of  $\{Q_k\}_{k>0}$ , because  $Y_1 = X_{Q_0}$ . Thus by the strong Markov property,

$$\mathbb{E}_i \left[ \prod_{k=1}^{W-1} \exp(\lambda Q_k) \middle| Q_0 \right] \leq \max_{j \in B} \mathbb{E} \left[ \prod_{k=1}^{W-1} \exp(\lambda Q_k) \middle| Y_1 = j \right],$$

so that

$$\mathbb{E}_i \left[ \prod_{k=0}^{W-1} \exp(\lambda Q_k) \right] \leq \mathbb{E}_i[\exp(\lambda Q_0)] \max_{j \in B} \mathbb{E} \left[ \prod_{k=1}^{W-1} \exp(\lambda Q_k) \middle| Y_1 = j \right].$$

We iteratively apply conditioning to show that

$$\mathbb{E}_i \left[ \prod_{k=0}^{W-1} \exp(\lambda Q_k) \right] \leq \mathbb{E}_i[\exp(\lambda Q_0)] \prod_{k=1}^{W-1} \left( \max_{j \in B} \mathbb{E}[\exp(\lambda Q_k) | Y_k = j] \right).$$

We can upper bound  $Q_k$  by assuming that it always goes on an excursion from  $B$ , such that

$$Q_k \leq 1 + (\text{length of an excursion into } B^c).$$

We invoke Hajek's result of Theorem 13, with  $V(x) < b + \nu_{\max}$  to bound the exponential moments of the excursion. For any  $i \in B$ ,

$$\mathbb{E}_i \left[ \prod_{k=0}^{W-1} \exp(\lambda Q_k) \right] \leq \left( e^\lambda \left( e^{\eta \nu_{\max}} \left( \frac{e^\lambda - 1}{1 - \rho e^\lambda} \right) + 1 \right) \right)^W,$$

where  $\eta$  and  $\rho$  are functions of  $\gamma$  and  $\nu_{\max}$ , as defined in (44) in Appendix F. They satisfy the conditions given in Hajek (1982). For  $\lambda < \min\left(0.43, \frac{0.8(1-\rho)}{\rho}\right)$ ,

$$e^\lambda < 1 + 1.25\lambda \quad \text{and} \quad 1 - \rho e^\lambda > 1 - \rho - 1.25\rho\lambda.$$

By substituting in these approximations and using  $1 + x < e^x$ , we obtain

$$\begin{aligned} \left( e^\lambda \left( e^{\eta \nu_{\max}} \left( \frac{e^\lambda - 1}{1 - \rho e^\lambda} \right) + 1 \right) \right)^W &< \left( e^\lambda \left( e^{\eta \nu_{\max}} \left( \frac{1.25\lambda}{1 - \rho - 1.25\rho\lambda} \right) + 1 \right) \right)^W \\ &< \exp(\lambda W) \exp\left(\frac{1.25\lambda W e^{\eta \nu_{\max}}}{1 - \rho - 1.25\rho\lambda}\right) \\ &< \exp\left(\lambda W \left( \frac{1.25 e^{\eta \nu_{\max}}}{1 - \rho - 1.25\rho\lambda} + 1 \right)\right). \end{aligned}$$

By Markov's inequality,

$$\begin{aligned} \mathbb{P}_i \left( \sum_{k=0}^{W-1} Q_k > Z \right) &\leq \frac{\mathbb{E}_i \left[ \exp\left(\lambda \sum_{k=0}^{W-1} Q_k\right) \right]}{\exp(\lambda Z)} \\ &\leq \exp\left(\lambda W \left( \frac{1.25 e^{\eta \nu_{\max}}}{1 - \rho - 1.25\rho\lambda} + 1 \right) - \lambda Z\right). \end{aligned}$$

Choose  $\lambda = \frac{0.8(1-\rho)}{e^{\eta \nu_{\max}}}$ . We can verify that for our choice of  $\eta$  and  $\rho$  according to (44),  $\lambda < \max\left(0.43, \frac{0.8(1-\rho)}{\rho}\right)$  always holds. Therefore, we complete the proof by substituting in for  $\lambda$ ,

$$\mathbb{P}_i \left( \sum_{k=0}^{W-1} Q_k > Z \right) \leq \exp\left(\frac{0.8(1-\rho)}{e^{\eta \nu_{\max}}} \left( \frac{1.25W e^{2\eta \nu_{\max}}}{(1-\rho)(e^{\eta \nu_{\max}} - \rho)} + W - Z \right)\right).$$

□

*Proof of Lemma 13.* By (40), for any constants  $W, Z \in \mathbb{Z}_+$ ,

$$\{T_i^B \leq W\} \cap \left\{ \sum_{k=0}^{W-1} Q_k \leq Z \right\} \implies \{T_i \leq Z\}.$$

We use the union bound on the contrapositive statement to obtain the inequalities

$$\begin{aligned} \mathbb{P}_i(T_i > Z) &\leq \mathbb{P}_i \left( \{T_i^B > W\} \cup \left\{ \sum_{k=0}^{W-1} Q_k > Z \right\} \right) \\ &\leq \mathbb{P}_i(T_i^B > W) + \mathbb{P}_i \left( \sum_{k=0}^{W-1} Q_k > Z \right). \end{aligned} \quad (41)$$

Choose  $W = 2H_i^B \left(2 + \frac{k}{R_i}\right)$  and

$$Z = 4H_i^B \left( \frac{1.25e^{2\eta\nu_{\max}}}{(1-\rho)(e^{\eta\nu_{\max}} - \rho)} + 1 \right) + \frac{\ln(2)e^{\eta\nu_{\max}}}{0.8(1-\rho)} + k = 2R_i - \frac{\ln(2)e^{\eta\nu_{\max}}}{0.8(1-\rho)} + k.$$

The next inequality follows from substituting these expressions for  $W$  and  $Z$  into (41), and applying Lemmas 7 and 15:

$$\begin{aligned} \mathbb{P}_i \left( T_i > 2R_i - \frac{\ln(2)e^{\eta\nu_{\max}}}{0.8(1-\rho)} + k \right) &\leq 2^{-\frac{k}{R_i}}, \\ \mathbb{P}_i(T_i > 2R_i + k) &\leq \mathbb{P}_i \left( T_i > 2R_i - \frac{\ln(2)e^{\eta\nu_{\max}}}{0.8(1-\rho)} + k \right) \leq 2^{-\frac{k}{R_i}}, \\ \mathbb{P}_i(T_i > k) &\leq 2^{-\frac{k-2R_i}{R_i}} \leq 4 \cdot 2^{-\frac{k}{R_i}}. \end{aligned}$$

□

*Proof of Theorem 7.* This proof follows a similar proof of Theorem 2. Substitute Lemma 13 into (8) to show that

$$1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{\mathbb{E}_i[T_i]} = \frac{1}{\mathbb{E}_i[T_i]} \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k) \leq \pi_i \left( \frac{4 \cdot 2^{-\theta^{(t)}/R_i}}{1 - 2^{-1/R_i}} \right). \quad (42)$$

Substitute (42) into (27) to complete the proof. □

*Proof of Theorem 8.* This proof follows a similar proof of Theorem 3. By dividing (8) by  $(1 - \mathbb{P}_i(T_i > \theta^{(t)}))$ , it follows that

$$\begin{aligned} \left| 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} \right| &= \left| \frac{1}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))} \left( \pi_i \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k) - \mathbb{P}_i(T_i > \theta^{(t)}) \right) \right| \\ &\leq \frac{1}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))} \max \left( \pi_i \sum_{k=\theta^{(t)}}^{\infty} \mathbb{P}_i(T_i > k), \mathbb{P}_i(T_i > \theta^{(t)}) \right). \end{aligned}$$

Then we apply Lemma 13 and use the fact that  $\mathbb{P}_i(T_i > \theta^{(t)}) < \frac{1}{2}$  to show that

$$\begin{aligned} \left| 1 - \frac{\mathbb{E}_i[\hat{T}_i^{(t)}]}{(1 - \mathbb{P}_i(T_i > \theta^{(t)}))\mathbb{E}_i[T_i]} \right| &\leq 2 \max \left( \pi_i \left( \frac{4 \cdot 2^{-\theta^{(t)}/R_i}}{1 - 2^{-1/R_i}} \right), 4 \cdot 2^{-\theta^{(t)}/R_i} \right) \\ &= 8 \cdot 2^{-\theta^{(t)}/R_i} \max \left( \frac{\pi_i}{1 - 2^{-1/R_i}}, 1 \right). \end{aligned} \quad (43)$$

Substitute (43) into (29) to complete the proof.  $\square$

*Proof of Theorem 9.* The proof is exactly the same as the proof of Theorem 6, except that we use Lemma 13 rather than Lemma 7.  $\square$

## Appendix E: Chernoff Bounds

**THEOREM 10 (Chernoff's Multiplicative Bound for Binomials).**

Let  $\{X_1, X_2, X_3, \dots, X_N\}$  be a sequence of independent identically distributed Bernoulli random variables, such that for all  $i$ ,  $X_i = 1$  with probability  $p$  and  $X_i = 0$  otherwise. Then for any  $\epsilon > 0$ ,

$$\mathbb{P} \left( \left| \frac{1}{N} \sum_{i=1}^N X_i - \mathbb{E}[X] \right| \geq \epsilon \mathbb{E}[X] \right) \leq 2e^{-\frac{\epsilon^2 N p}{3}}.$$

**THEOREM 11 (Chernoff's Multiplicative Bound for Bounded Variables).**

Let  $\{X_1, X_2, X_3, \dots, X_N\}$  be a sequence of independent identically distributed strictly bounded non-negative random variables, such that  $X_i \sim X$  for all  $i$ , and  $X \in [0, \theta]$ . Then for any  $\epsilon > 0$ ,

$$\mathbb{P} \left( \left| \frac{1}{N} \sum_{i=1}^N X_i - \mathbb{E}[X] \right| \geq \epsilon \mathbb{E}[X] \right) \leq 2e^{-\frac{\epsilon^2 N \mathbb{E}[X]}{3\theta}}.$$

## Appendix F: Lyapunov Function Analysis

**THEOREM 12 (Foster 1953).** Let  $\{X_t\}$  be a discrete time, irreducible Markov chain on countable state space  $\Sigma$  with transition probability matrix  $P$ .  $\{X_t\}$  is positive recurrent if and only if there exists a Lyapunov function  $V : \Sigma \rightarrow \mathbb{R}_+$ ,  $\gamma > 0$  and  $b \geq 0$ , such that

1. For all  $x \in \Sigma$ ,

$$\mathbb{E}[V(X_{t+1}) | X_t = x] \leq \infty,$$

2. For all  $x \in \Sigma$  such that  $V(x) > b$ ,

$$\mathbb{E}[V(X_{t+1}) - V(X_t) | X_t = x] \leq -\gamma.$$

In words, given a positive recurrent Markov chain, there exists a Lyapunov function  $V : \Sigma \rightarrow \mathbb{R}_+$  and a decomposition of the state space into  $B = \{x \in \Sigma : V(x) \leq b\}$  and  $B^c = \{x \in \Sigma : V(x) > b\}$  such that there is a uniform negative drift in  $B^c$  towards  $B$  and  $|B|$  is finite.

For any irreducible, Markov chain, the following function is a valid Lyapunov function for  $\gamma = 1$  and  $b = 0.5$ : Choose any node  $i \in \Sigma$ , and fix this as the ‘‘central node’’. Define the function  $V : \Sigma \rightarrow$

$\mathbb{R}_+$  such that  $V(i) = 0$  and for all  $j \in \Sigma \setminus \{i\}$ ,  $V(j) = \mathbb{E}_j[T_i]$ . By definition,  $B = \{i\}$ . For all  $x \in \Sigma$ , by positive recurrence,  $\mathbb{E}[V(X_{t+1})|X_t = x] \leq \infty$ . Similarly, for all  $x$  such that  $V(x) > b$ ,

$$V(x) = \mathbb{E}_x[T_i] = 1 + \sum_{y \in \Sigma} P_{xy} E_y[T_i] = 1 + \mathbb{E}[V(X_{t+1})|X_t = x].$$

Therefore, for all  $x \in B^c$ ,

$$\mathbb{E}[V(X_{t+1}) - V(X_t)|X_t = x] = -1 \leq -\gamma.$$

**THEOREM 13 (Hajek 1982).** *Let  $\{X_t\}$  be an irreducible, positive recurrent Markov chain on a countable state space  $\Sigma$  with transition probability matrix  $P$ . Assume that there exists a Lyapunov function  $V : \Sigma \rightarrow \mathbb{R}_+$  and values  $\nu_{\max}, \gamma > 0$ , and  $b \geq 0$  satisfying Assumption 1. Let the random variable  $\tau_B = \inf\{t : X_t \in B\}$ . Then for any  $x$  such that  $V(x) > b$ , and for any choice of constants  $\omega > 0$ ,  $\eta$ ,  $\rho$ , and  $\lambda$  satisfying*

$$\begin{aligned} 0 < \eta &\leq \min\left(\omega, \frac{\gamma\omega^2}{e^{\omega\nu_{\max}} - (1 + \omega\nu_{\max})}\right), \\ \rho &= 1 - \gamma\eta + \frac{(e^{\omega\nu_{\max}} - (1 + \omega\nu_{\max}))\eta^2}{\omega^2}, \\ &\text{and } 0 < \lambda < \ln\left(\frac{1}{\rho}\right), \end{aligned}$$

the following two inequalities hold:

$$\begin{aligned} \mathbb{P}[\tau_B > k | X_0 = x] &\leq e^{\eta(V(x)-b)} \rho^k, \\ \text{and } \mathbb{E}[e^{\lambda\tau_B} | X_0 = x] &\leq e^{\eta(V(x)-b)} \left(\frac{e^\lambda - 1}{1 - \rho e^\lambda}\right) + 1. \end{aligned}$$

A concrete set of constants that satisfy the conditions above are

$$\omega = \frac{1}{\nu_{\max}}, \eta = \frac{\gamma}{2(e-2)\nu_{\max}^2}, \text{ and } \rho = 1 - \frac{\gamma^2}{4(e-2)\nu_{\max}^2}. \quad (44)$$

## Endnotes

1. A short version of this paper will appear in the Neural Information Processing Systems Conference Proceedings (NIPS, Dec 2013).
2. We denote  $\tilde{O}(f(a)g(b)) = \tilde{O}(f(a)\text{polylog}f(a))\tilde{O}(g(b)\text{polylog}g(b))$ .
3.  $\mathbf{1}$  denotes the all ones vector.
4.  $e_i$  denotes the standard basis vector having value one in coordinate  $i$  and zero for all other coordinates.
5. Throughout the paper, *Markov chain* and *random walk* on a graph are used interchangeably; similarly *nodes* and *states* are used interchangeably.

6. A Markov chain is irreducible if and only if the corresponding Markov chain graph is strongly connected, i.e. for all  $x, y \in \Sigma$ , there exists a path from  $x$  to  $y$ .

7. A Markov chain is positive recurrent if the expected time for a random walk beginning at node  $i$  to return to node  $i$  is finite. This means that the random walk cannot “drift to infinity”. This is true for all irreducible finite state space Markov chains.

8. In this example, even if we did not limit the algorithm to using a local neighborhood of  $i$ , the probability that a random walk beginning at node 1 reaches node  $n_1$  before returning to state 1 is very small, decaying exponentially in  $n_1$ . Therefore, we would need an exponential in  $n_1$  number of samples in order for there to be a reasonable probability that one of the sample random walks explored the region past node  $n_1$ . This presents another fundamental limitation of computing stationary distributions for Markov chains that are not well connected.

## Acknowledgments

This work is supported in parts by ARO under MURI awards 58153-MA-MUR and W911NF-11-1-0036, and grant 56549-NS, and by NSF under grant CIF 1217043 and a Graduate Fellowship.

## References

- Aldous, D., J. Fill. 1999. Reversible Markov chains and random walks on graphs: Chapter 2 (General Markov chains). book in preparation. URL: <http://www.stat.berkeley.edu/~aldous/RWG/Chap2.pdf> 7, 19–20.
- Andersen, R., C. Borgs, J. Chayes, J. Hopcraft, V.S. Mirrokni, S.H. Teng. 2007. Local computation of PageRank contributions. *Proceedings of the 5th international conference on Algorithms and models for the web-graph*. WAW’07, Springer-Verlag, Berlin, Heidelberg, 150–165.
- Assmussen, S., P. Glynn. 2010. *Stochastic Simulation: Algorithms and Analysis (Stochastic Modeling and Applied Probability)*. Springer.
- Avrachenkov, K., N. Litvak, D. Nemirovsky, N. Osipova. 2007. Monte Carlo methods in PageRank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis* **45**(2) 890–904.
- Bahmani, B., A. Chowdhury, A. Goel. 2010. Fast incremental and personalized PageRank. *Proc. VLDB Endow.* **4**(3) 173–184.
- Bertsimas, D., D. Gamarnik, J.N. Tsitsiklis. 1998. Geometric bounds for stationary distributions of infinite Markov chains via Lyapunov functions. Tech. rep., MIT Sloan School of Management.
- Borgs, C., M. Brautbar, J. Chayes, S.H. Teng. 2012. Sublinear time algorithm for PageRank computations and related applications. *CoRR* **abs/1202.2771**.
- Candogan, O., K. Bimpikis, A. Ozdaglar. 2012. Optimal pricing in networks with externalities. *Operations Research* **60**(4) 883–905.
- Chasparis, G.C., J.S. Shamma. 2010. Control of preferences in social networks. *CDC*. 6651–6656.

- Chen, Y.Y., Q. Gan, T. Suel. 2004. Local methods for estimating PageRank values. *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 381–389.
- Cipra, B. 2000. The best of the 20th century: Editors name top 10 algorithms. *SIAM News* **33**(4) 1.
- Diaconis, P. 2009. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society* **46**(2) 179–205.
- Diaconis, P., L. Saloff-Coste. 1998. What do we know about the Metropolis algorithm? *Journal of Computer and System Sciences* **57**(1) 20–36.
- Fogaras, D., B. Racz, K. Csalogany, T. Sarlos. 2005. Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments. *Internet Mathematics* **2**(3) 333–358.
- Golub, G.H., C.F. Van Loan. 1996. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press.
- Hajek, B. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability* 502–525.
- Hastings, W.K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1) 97–109.
- Haveliwala, T.H. 2003. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. Technical Report 2003-29, Stanford InfoLab. Extended version of the WWW 2002 paper on Topic-Sensitive PageRank.
- Jeh, G., J. Widom. 2003. Scaling personalized web search. *Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA, 271–279.
- Kamvar, S., T. Haveliwala, C. Manning, G. Golub. 2003. Exploiting the block structure of the web for computing PageRank. *Stanford University Technical Report* .
- Koury, J.R., D.F. McAllister, W.J. Stewart. 1984. Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains. *SIAM Journal on Algebraic Discrete Methods* **5**(2) 164–186.
- Levin, D.A., Y. Peres, E.L. Wilmer. 2009. *Markov chains and mixing times*. Amer Mathematical Society.
- Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21** 1087.
- Meyn, S.P., R.L. Tweedie. 1993. *Markov chains and stochastic stability*. Springer-Verlag.
- Negahban, S., S. Oh, D. Shah. 2012. Iterative ranking from pair-wise comparisons. *CoRR* **abs/1209.1688**.
- Newman, M. E. J. 2010. *Networks: an introduction*. Oxford University Press, Oxford; New York.
- Page, L., S. Brin, R. Motwani, T. Winograd. 1999. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66.



- Sarma, A. Das, S. Gollapudi, R. Panigrahy. 2011. Estimating PageRank on graph streams. *Journal of ACM* **58**(3) 13.
- Sarma, A. Das, A.R. Molla, G. Pandurangan, E. Upfal. 2012. Fast distributed PageRank computation. *CoRR* abs/1208.3071.
- Semkow, T.M., S. Pomm, S. Jerome, D.J. Strom, eds. 2006. *Applied Modeling and Computations in Nuclear Science*. American Chemical Society, Washington, DC.
- Shah, D., T. Zaman. 2011. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory* **57**(8) 5163–5181.
- Stewart, W.J. 1994. *Introduction to the numerical solution of Markov chains*, vol. 41. Princeton University Press Princeton.