Non-Markovian Monte Carlo on Directed Graphs

CHUL-HO LEE, Florida Institute of Technology
MIN KANG, North Carolina State University
DO YOUNG EUN, North Carolina State University

Markov Chain Monte Carlo (MCMC) has been the de facto technique for sampling and inference of large graphs such as online social networks. At the heart of MCMC lies the ability to construct an ergodic Markov chain that attains any given stationary distribution $\boldsymbol{\pi}$, often in the form of random walks or crawling agents on the graph. Most of the works around MCMC, however, presume that the graph is undirected or has reciprocal edges, and become inapplicable when the graph is directed and non-reciprocal. Here we develop a similar framework for directed graphs, which we call Non-Markovian Monte Carlo (NMMC), by establishing a mapping to convert $\boldsymbol{\pi}$ into the quasi-stationary distribution of a carefully constructed transient Markov chain on an extended state space. As applications, we demonstrate how to achieve any given distribution $\boldsymbol{\pi}$ on a directed graph and estimate the eigenvector centrality using a set of non-Markovian, history-dependent random walks on the same graph in a distributed manner. We also provide numerical results on various real-world directed graphs to confirm our theoretical findings, and present several practical enhancements to make our NMMC method ready for practical use in most directed graphs. To the best of our knowledge, the proposed NMMC framework for directed graphs is the first of its kind, unlocking all the limitations set by the standard MCMC methods for undirected graphs.

CCS Concepts: • **Mathematics of computing** → **Probabilistic algorithms**; **Stochastic processes**;

Additional Key Words and Phrases: Directed graphs, sampling, eigenvector centrality, transient Markov chains, non-reversible Markov chains, quasi-stationary distribution, history-dependent random walks

ACM Reference Format:
Chul-Ho Lee, Min Kang, and Do Young Eun. 2019. Non-Markovian Monte Carlo on Directed Graphs. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 1 (January 2019), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Markov Chain Monte Carlo (MCMC) is a general technique for sampling from an arbitrary probability distribution by constructing a Markov chain that achieves the target distribution as its stationary distribution [47]. By definition, every MCMC method requires the so-called balance condition, i.e., $\pi(j) = \sum_i \pi(i)P_{ij}$ for all j, where P_{ij} is the transition probability from state i to j and $\boldsymbol{\pi} = [\pi(1), \pi(2), \ldots, \pi(n)]$ is the stationary distribution. Most popular MCMC methods such as Metropolis-Hastings algorithm and Gibbs sampler, often referred to as Glauber dynamics, however, require a simple yet more restrictive condition, called *detailed* balance condition, i.e., $\pi(i)P_{ij} = \pi(j)P_{ji}$ for all i, j [38, 47, 56]. The detailed balance condition is generally preserved under a proposal and acceptance framework for state transitions. Assuming the current state i, a transition

Authors' addresses: Chul-Ho Lee, Department of Computer Engineering and Sciences, Florida Institute of Technology, Melbourne, FL, 32901, clee@fit.edu; Min Kang, Department of Mathematics, North Carolina State University, Raleigh, NC, 27695, mkang2@ncsu.edu; Do Young Eun, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 27695, dyeun@ncsu.edu.

2476-1249/2019/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to state j is proposed with probability Q_{ij}, which is the transition probability of an irreducible (proposal) Markov chain, and then accepted with acceptance probability γ_{ij}. This two-step process ensures the convergence to the desired stationary distribution $\boldsymbol{\pi}$ via achieving the detailed balance condition, i.e., $\pi(i)Q_{ij}\gamma_{ij} = \pi(j)Q_{ji}\gamma_{ji}$ for all i, j [1]. The detailed balance condition imposes a serious restriction, named 'reciprocity' condition, that $Q_{ij} > 0$ if and only if $Q_{ji} > 0$ [26, 47].

The detailed balance condition is a local condition that eases the construction of a Markov chain in MCMC methods, but is also criticized as a source of slow convergence to equilibrium and high sampling variability because every transition should be balanced out with its reverse transition. Thus, some recent advances in MCMC methods have been made to remove the requirement of the detailed balance condition in order to accelerate the convergence speed or improve sampling efficiency. A popular class of strategies is to transform a given reversible Markov chain to its related nonreversible chain by using the idea of lifting [22, 24, 29, 62, 63, 68, 74], nonbacktracking [2, 25, 51], or vorticity [12, 23, 41, 42, 57, 66]. The former two ideas are based on expanding the state space to give rise to persistence of motion or directionality in state transitions. The other idea is that for the transition matrix of a given reversible chain, which is symmetric with respect to $\boldsymbol{\pi}$, a proper antisymmetric matrix with $\boldsymbol{\pi}$ (or more matrices) is added to introduce a non-trivial probability flux flowing over a cycle in one direction. They all, however, require the reciprocity condition, which is carried over from the original reversible chain before the transformation.

On the other hand, MCMC also fits into the problem of sampling from a target probability distribution over a large, *unknown* graph, or a huge state space whose exploration can only be done by traversing along the edges between nodes (states). The sampling problem with such limited access to the space often arises when sampling and estimating structural properties of various types of large networks, including peer-to-peer networks and online social networks, whose complete information is generally *unavailable* in advance. MCMC has then been a key enabler to obtain a sequence of samples by constructing an ergodic Markov chain or 'random walk' on a given graph, whose equilibrium distribution equals the target distribution, with only a limited and *local* knowledge of the graph structure [5, 32, 37, 44, 45, 58, 65, 76–79].

Most of the MCMC methods for constructing random walks on graphs, however, become *inapplicable to directed* graphs, though many real-world networks can still only be represented by directed graphs [28, 52, 53, 67]. The famous examples are Web graphs, online social networks having follower-followee relationship, neuronal and metabolic networks, to name a few. The directed graphs naturally violate the reciprocity condition, i.e., there is at least one directed edge (i, j) for which $Q_{ij} > 0$ but $Q_{ji} = 0$ due to the absence of its reverse directed edge (j, i). The lack of reciprocity condition simply prohibits us from using the usual MCMC methods and even obtaining a closed-form expression of the stationary distribution of a given Markov chain on a directed graph. Thus, the stationary distribution has been merely estimated based on the visit counts obtained by running the Markov chain on a directed graph without knowing the explicit form of the distribution [6, 40, 61]. This is also the very reason why popular social-network-analysis metrics defined on a directed graph such as PageRank and eigenvector centrality are generally computed and estimated by the power method, assuming that a global knowledge of the whole graph structure is available a priori [53].

[1] While the proposal and acceptance framework is mostly used for the Metropolis-Hastings algorithm, (random scan) Gibbs sampler also fits into the framework where the conditional distribution of the Gibbs sampler is considered as the proposal distribution along with the acceptance probability of one. In this sense, the Gibbs sampler is a special case of the Metropolis-Hastings algorithm. Furthermore, when it comes to binary spin systems, it is essentially the Barker's method [8] that was compared with the Metropolis algorithm under the proposal and acceptance framework in [38, 56].

In this paper, we present a general algorithmic framework called Non-Markovian Monte Carlo (NMMC) that constructs a set of non-Markovian random walks whose historical empirical distribution converges to any desired probability distribution $\boldsymbol{\pi}$ over a directed, non-reciprocal graph, using only local information on the graph structure. The main idea is to establish a non-trivial mapping to convert the desired distribution into the so-called *quasi-stationary distribution* (QSD) of a duly constructed *transient* Markov chain on an extended state space. We then leverage history-dependent random walks or reinforced walks to estimate the QSD "on the fly", which does not require the entire topological information. This novel construction leads to a large degree of freedom and does not require the reciprocity condition that has been implicitly or explicitly assumed in almost all MCMC methods, and thus makes our approach the only viable solution to directed graphs. As a byproduct, our NMMC framework also allows us to estimate the eigenvector centrality of a given directed graph, one of the key metrics used in ranking or identifying influential nodes in epidemics or information spreading, by using the same set of history-dependent random-walk crawling agents.

The rest of this paper is structured as follows. In Section 2, we collect several terms and definitions on nonreversible Markov chains on directed graphs as well as on the eigenvector centrality (EVC) of a directed graph, to be used throughout the paper. In Section 3, we provide necessary background on the QSD, its key properties and characterizations, and then formally establish a mapping to convert a given distribution $\boldsymbol{\pi}$ and the EVC of a graph into the QSD of a carefully constructed transient Markov chain. In Section 4, we first explain recent developments in the literature to estimate the QSD and then build our NMMC method with theoretical guarantees to achieve a given $\boldsymbol{\pi}$ and to estimate the EVC using random crawlers on directed graphs. We also discuss several ways to improve the speed of convergence and the practicability including dynamically adjusted algorithmic parameters based on run-time estimates on the fly. Section 5 presents our extensive simulation results using a set of real-life directed graphs to numerically support our theoretical findings. In Section 6, we thoroughly compare our NMMC method to the relevant literature, ranging from the traditional MCMC methods to the PageRank algorithm. We also present how to relax (potentially impractical) assumptions with our NMMC method, which require the in-degree information and the strong connectedness of the underlying directed graph, and demonstrate the feasibility of our NMMC method as a truly working solution to most directed graphs in practice. We provide related prior studies around sampling of directed graphs in Section 7 and conclude with a brief summary capturing our novel contributions in Section 8.

2 PRELIMINARIES

2.1 Nonreversible Markov Chains on Directed Graphs

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a finite, aperiodic, strongly connected graph with a set of nodes $\mathcal{N} = \{1, 2, \ldots, n\}$ and a set of *directed* edges $\mathcal{E}$. The graph $\mathcal{G}$ is defined by an $n \times n$ adjacency matrix $\mathbf{A}$, with elements $A_{ij} = 1$ if there is an edge from node i to j (directed), i.e., $(i, j) \in E$, and $A_{ij} = 0$ otherwise. Here $\mathbf{A}$ is asymmetric, namely, there is at least one directed edge (i, j) for which $A_{ij} = 1$ but $A_{ji} = 0$ (non-reciprocal). Let $d_i^+ = \sum_{j \in \mathcal{N}} A_{ij}$ be the out-degree of node i, i.e., the number of out-bound edges emanating from i. Similarly, let $d_i^- = \sum_{j \in \mathcal{N}} A_{ji}$ be the in-degree of node i (the number of edges pointing to node i). We shall demonstrate how the 'strong connectedness' of the underlying graph $\mathcal{G}$ can be relaxed in Section 6.4.

Consider an ergodic Markov chain on $\mathcal{N}$ with its transition matrix $\mathbf{Q} = [Q_{ij}]$ such that $Q_{ij} > 0$ for every $(i, j) \in \mathcal{E}$, or $A_{ij} = 1$, which is irreducible and aperiodic on $\mathcal{N}$. The irreducibility of $\mathbf{Q}$ is granted from the connectivity of the graph $\mathcal{G}$, or the irreducibility of the adjacency matrix $\mathbf{A}$, in that for any $i, j \in \mathcal{N}$, there exists a path following directed edges from i to j, i.e., there exists

$k > 0$ with $[\mathbf{A}^k]_{ij} > 0$. Similarly for the aperiodicity of $\mathbf{Q}$. Then, since $\mathbf{A}$ is asymmetric, the ergodic Markov chain is nonreversible without satisfying the reciprocity condition. Its unique stationary distribution $\boldsymbol{\mu} = [\mu(1), \mu(2), \cdots, \mu(n)]$, which satisfies $\mu(j) = \sum_{i \in \mathcal{N}} \mu(i) Q_{ij}$ for all $j \in \mathcal{N}$, is also simply *unknown* for a general graph $\mathcal{G}$. For example, if we define a simple random walk on $\mathcal{G}$, it is an ergodic, nonreversible Markov chain on $\mathcal{N}$ with its transition matrix $\mathbf{Q}^{\mathsf{srw}} = [Q^{\mathsf{srw}}_{ij}]$ given by $Q^{\mathsf{srw}}_{ij} = A_{ij}/d^+_i$, where $d^+_i > 0$ for all i. There is, however, no closed-form expression of the stationary distribution $\boldsymbol{\mu}$ for the simple random walk on a general directed graph.

2.2 Eigenvector Centrality

In addition to sampling from a target probability distribution $\boldsymbol{\pi}$ on a directed graph, another metric of interest to estimate on the directed graph is the so-called eigenvector centrality (EVC). Specifically, the EVC of node $i \in \mathcal{N}$, say $x(i)$, represents the notion of importance (centrality) of node i and is proportional to the sum of the EVC values of nodes pointing to i, i.e.,

$$\sum_{i \in \mathcal{N}} x(i) A_{ij} = \lambda x(j) \ \text{ for all } j \in \mathcal{N} \iff \mathbf{x}\mathbf{A} = \lambda \mathbf{x}, \tag{1}$$

with some positive constant $\lambda > 0$, where $\mathbf{x}$ is a row vector $\mathbf{x} = [x(1), x(2), \cdots, x(n)]$. Since the adjacency matrix $\mathbf{A}$ is irreducible, the Perron-Frobenius theorem [50] asserts that there exists a unique solution $\mathbf{x}$ to (1), which is the left leading eigenvector of $\mathbf{A}$ with all positive entries $x(i) > 0$, $i = 1, 2, \ldots, n$, corresponding to the largest eigenvalue $\lambda = \lambda(\mathbf{A}) > 0$, i.e., the spectral radius of $\mathbf{A}$. Throughout this paper, we refer to this left leading eigenvector $\mathbf{x}$ as the EVC of $\mathbf{A}$. Since $\beta\mathbf{x}$ ($\beta \neq 0$) can also be a leading eigenvector for $\lambda(\mathbf{A})$, we also assume that $\mathbf{x}$ is normalized such that $\sum_{i \in \mathcal{N}} x(i) = 1$.

The EVC finds many important applications. It has been playing a crucial role in ranking nodes in social network analysis and identifying influential nodes (users) in epidemics or information spreading [18, 19, 28, 48, 53], and the reciprocal of its corresponding eigenvalue $1/\lambda(\mathbf{A})$ is known to be the epidemic threshold of a general graph for SIR/SIS epidemic processes [20, 21, 27, 31, 72]. Again, we note that there is no closed-form expression of the EVC $\mathbf{x}$. While the EVC has also been used as a main kernel for the maximal entropy random walk in the statistical physics [17, 64], its practical construction based only on local information still remains an open issue.

Even worse, unlike the stationary distribution $\boldsymbol{\mu}$ of an ergodic Markov chain, e.g., the simple random walk, satisfying $\boldsymbol{\mu} = \boldsymbol{\mu}\mathbf{Q}$ with its (stochastic) transition matrix $\mathbf{Q}$, the EVC $\mathbf{x}$ in (1) cannot be made as a stationary distribution of an ergodic Markov chain (or a random walk) unless the adjacency matrix $\mathbf{A}$ can be made stochastic by some constant multiple, i.e., $d^+_i = d$ for all i. Even if this is the case, the solution $\mathbf{x}$ to (1) defies any closed-form expression since the in-degrees d^-_i can be all different, reducing to the same situation with an ergodic Markov chain on a directed graph whose stationary distribution $\boldsymbol{\mu}$ is unknown. Thus, the only viable approach to compute the EVC $\mathbf{x}$ is again to rely on the power method for which the entire adjacency matrix $\mathbf{A}$ needs to be known a priori. To the best of our knowledge, it remains open or unknown as to how (or whether or not it is possible) to build crawling algorithms using only local information towards estimating the EVC $\mathbf{x}$.

3 NMMC: MAPPING TO QSD

The main goal of our proposed NMMC framework is to achieve any arbitrary probability distribution $\boldsymbol{\pi}$ on a general directed graph $\mathcal{G}$ and also to estimate the EVC of its adjacency matrix $\mathbf{A}$ using a set of random walks in a distributed manner without requiring the entire matrix $\mathbf{A}$ a priori. The main enabler of our NMMC framework to tackle this challenge is to convert any given ergodic, nonreversible Markov chain (e.g., simple random walk) on $\mathcal{N}$ with $\mathbf{Q} = [Q_{ij}]_{i,j \in \mathcal{N}}$ and *unknown* $\boldsymbol{\mu}$

into a related suitable *transient* Markov chain on an augmented state space $\mathcal{N}^+ := \mathcal{N} \cup \{0\}$, where the node set $\mathcal{N}$ is a set of transient (non-absorbing) states and '0' is an (artificially added) absorbing state. This mapping is made based solely on local information in a way that the resulting *quasi-stationary distribution* (QSD) becomes the target distribution $\boldsymbol{\pi}$. Later in Section 4 we present how to efficiently estimate the required QSD using a set of non-Markovian, history-dependent random walks, which is yet another integral part of our NMMC framework. In what follows, we first collect several definitions and basic equations we need for later use.

3.1 A Primer on QSD

Suppose that we have a discrete time *transient* Markov chain $\{X_t\}_{t\geq 0}$ on a finite state space $\mathcal{N}^+ := \mathcal{N} \cup \{0\}$ with its transition matrix $\mathbf{P} = [P_{ij}]_{i,j\in N^+}$, where $\mathcal{N} = \{1, 2, \cdots, n\}$ is the set of transient (non-absorbing) states and '0' is the absorbing state with $P_{00} = 1$, i.e., once the chain hits 0, it remains there forever. We have $P_{i0} > 0$ for at least one $i \in \mathcal{N}$ to ensure that the chain is transient. Clearly, $\mathbf{P}$ is a stochastic matrix on $\mathcal{N}^+$. We assume that this Markov chain is irreducible and aperiodic on the set of *non-absorbing* states in the sense that $\mathbb{P}\{X_t = j \text{ for some } t \geq 1 \,|X_0 = i\} > 0$ and g.c.d.$\{t > 0 \,:\, P^t_{ii} > 0\} = 1$ for all $i, j \in \mathcal{N}$, where $P^t_{ii} = \mathbb{P}\{X_t = i \,|X_0 = i\}$ is a t-step transition probability.[2] Then, we have $\lim_{t\to\infty} X_t = 0$ almost surely (i.e., absorption is certain eventually). If we restrict the stochastic matrix $\mathbf{P}$ only on $\mathcal{N}$, the resulting $n \times n$ matrix $\tilde{\mathbf{P}} = [\tilde{P}_{ij}]_{i,j\in\mathcal{N}}$ is naturally a *sub-stochastic* matrix.

Consider the dynamics of the transient Markov chain $\{X_t\}$ at any given time t by conditioning on the event that the chain is still surviving, i.e., the chain has not hit the absorbing site '0' by time t. These conditional probabilities $\mathbb{P}\{X_t \in \cdot \,|X_t \in \mathcal{N}\} = \mathbb{P}\{X_t \in \cdot \,|X_t \neq 0\}$ are well-defined time-evolving full probability measures fully supported on the set of non-absorbing states $\mathcal{N}$. Now, we can consider the stationary probability measure with respect to these conditional probability measures; in other words, the invariant distribution $\boldsymbol{\nu} = [\nu(1), \nu(2), \cdots, \nu(n)]$ under the Markovian dynamics *conditioned to survive*. This stationary distribution $\boldsymbol{\nu}$ is called the quasi-stationary distribution (QSD) [10, 13, 35, 36, 49, 70, 71], satisfying

$$\nu(j) = \mathbb{P}\{X_t = j \mid X_0 \sim \boldsymbol{\nu}, X_t \neq 0\} \tag{2}$$

for all non-absorbing states $j \in \mathcal{N}$ and for all time $t \geq 0$, where $X_0 \sim \boldsymbol{\nu}$ means that the initial state X_0 is drawn from the distribution $\boldsymbol{\nu}$.

Exactly in the same way as we approximate the stationary distribution of an ergodic Markov chain from its long-term limiting behavior, we can take a similar line of approach to acquire the desired QSD. Specifically, by taking the limit on the conditional distribution at time t in (2) starting from an initial distribution $\boldsymbol{\mu}_0$, we obtain

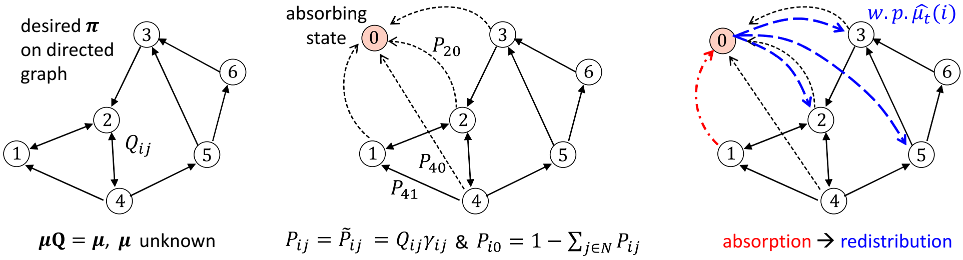
$$\lim_{t\to\infty} \mathbb{P}\{X_t = j \mid X_0 \sim \boldsymbol{\mu}_0, X_t \neq 0\} = \nu'(j) \tag{3}$$

for all $j \in \mathcal{N}$. If there exists an initial distribution $\boldsymbol{\mu}_0$ such that $\boldsymbol{\nu}' := [\nu'(1), \nu'(2), \cdots, \nu'(n)]$ exists, we call this limiting distribution a quasi limiting distribution (QLD). It then follows that under our finite state space setting with irreducible and aperiodic assumptions, the QSD $\boldsymbol{\nu}$ and the QLD $\boldsymbol{\nu}'$ coincide, i.e., $\boldsymbol{\nu} = \boldsymbol{\nu}'$, and they are uniquely determined for any initial distribution $\boldsymbol{\mu}_0$ [71].

We can rewrite (2) with $t = 1$ as, for any $j \in \mathcal{N}$,

$$\nu(j) = \frac{\mathbb{P}\{X_1 = j \mid X_0 \sim \boldsymbol{\nu}\}}{\mathbb{P}\{X_1 \neq 0 \mid X_0 \sim \boldsymbol{\nu}\}} = \frac{\sum_{i\in\mathcal{N}} \nu(i) P_{ij}}{1 - \sum_{i\in\mathcal{N}} \nu(i) P_{i0}},$$

[2] Equivalently, this is often termed as $\mathcal{N}$ being an irreducible class [10, 70, 71].


desired $\boldsymbol{\pi}$ on directed graph
Q_{ij}
$\boldsymbol{\mu}\mathbf{Q} = \boldsymbol{\mu}$, $\boldsymbol{\mu}$ unknown
absorbing state
P_{20}
P_{40}
P_{41}
$P_{ij} = \tilde{P}_{ij} = Q_{ij}\gamma_{ij}$ & $P_{i0} = 1 - \sum_{j \in N} P_{ij}$
$w.p. \hat{\mu}_t(i)$
absorption → redistribution
0
1
2
3
4
5
6

(a) Ergodic, nonreversible **Q** (b) Mapping $\boldsymbol{\pi}$ to QSD of transient chain **P** (c) Algorithm for QSD approximation

Fig. 1. Summary of our Non-Markovian Monte Carlo method: With any ergodic, nonreversible Markov chain $\mathbf{Q}$ defined over the node set $\mathcal{N}$ of a directed graph $\mathcal{G}$, we construct a transient Markov chain $\mathbf{P}$ on $\mathcal{N}^+ = \mathcal{N} \cup \{0\}$ such that its QSD becomes our desired distribution $\boldsymbol{\pi}$ and then leverage non-Markovian, history-dependent random walks to estimate the desired QSD $\boldsymbol{\pi}$.

which then reads as

$$\sum_{i \in \mathcal{N}} \nu(i) P_{ij} = \lambda \nu(j), \tag{4}$$

where

$$\lambda = \mathbb{P}\{X_1 \neq 0 \mid X_0 \sim \boldsymbol{\nu}\} = 1 - \sum_{i \in \mathcal{N}} \nu(i) P_{i0}. \tag{5}$$

In other words, since $P_{ij} = \tilde{P}_{ij}$ for $i, j \in \mathcal{N}$, we can rewrite (4) as $\boldsymbol{\nu}\tilde{\mathbf{P}} = \lambda\boldsymbol{\nu}$, where $\tilde{\mathbf{P}}$ is the sub-stochastic matrix, which is a restriction on $\mathcal{N}$ from the original transition matrix $\mathbf{P}$ on $\mathcal{N}^+$. Then, by the Perron-Frobenius theorem for irreducible, non-negative matrices (not necessarily stochastic) [3], the QSD $\boldsymbol{\nu}$ must be the uniquely determined left principal eigenvector of $\tilde{\mathbf{P}}$ with $\nu(j) > 0$ for all $j \in \mathcal{N}$, corresponding to the maximal eigenvalue $\lambda > 0$. Note that $1 - \lambda = \sum_{i \in \mathcal{N}} \nu(i) P_{i0} > 0$ is the one-step absorption probability starting from the QSD $\boldsymbol{\nu}$.

3.2 Achieving π via QSD

We next show how to construct a transient Markov chain $\mathbf{P} = [P_{ij}]$ on the augmented state space $\mathcal{N}^+ = \mathcal{N} \cup \{0\}$ with a virtual absorbing site '0' added to the original state space $\mathcal{N}$, such that the resulting QSD $\boldsymbol{\nu}$ becomes any desired target probability distribution $\boldsymbol{\pi} = [\pi(1), \pi(2), \cdots, \pi(n)]$ with $\pi(i) > 0$ for $i \in \mathcal{N}$, i.e., $\boldsymbol{\nu} = \boldsymbol{\pi}$. This mapping, as we shall show shortly, is always possible as long as the underlying nonreversible Markov chain $\mathbf{Q} = [Q_{ij}]$ on the original state space $\mathcal{N}$ (or the directed graph $\mathcal{G}$) is ergodic. Note that letting $\boldsymbol{\mu} = [\mu(1), \mu(2), \cdots, \mu(n)]$ be the stationary distribution of the ergodic chain $\mathbf{Q}$, satisfying $\boldsymbol{\mu} = \boldsymbol{\mu}\mathbf{Q}$, we do not require any knowledge on $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}$ is *unknown* in our construction, as illustrated in Figure 1(a).

Adopting the proposal and acceptance framework as in usual MCMC methods, we first use the transition probability Q_{ij} as a proposal distribution from the current state (node) i to j ($i, j \in \mathcal{N}$). This proposed move is then 'accepted' with probability γ_{ij}, which will be determined shortly. Unlike usual MCMC methods, however, with probability $1 - \gamma_{ij}$, this move is 'rejected', in which case the chain gets absorbed to '0' and stays there forever. We thus have $P_{ij} = Q_{ij}\gamma_{ij}$ for $i, j \in \mathcal{N}$, and $P_{i0} = \sum_{j \in \mathcal{N}} Q_{ij}(1 - \gamma_{ij})$ for each $i \in \mathcal{N}$, with $P_{00} = 1$. Recall that $\tilde{P}_{ij} = P_{ij}$ for $i, j \in \mathcal{N}$. See for Figure 1(b) for an illustration.

Despite the similar structure, our proposed method is fundamentally different from the MCMC methods, including the famous Metropolis-Hastings (MH) algorithm. The crucial difference is that

[3] The Perron-Frobenius theorem asserts that the leading (left) eigenvector of any irreducible, non-negative matrix corresponding to its largest eigenvalue is the only one with every component being positive [50].

our proposed chain $\mathbf{Q}$ is *nonreversible* on a directed graph *without the reciprocity condition*, and goes to the absorbing site '0' if rejected. In contrast, in the MH algorithm the proposed chain is on a undirected graph (typically reversible), and if rejected, it simply stays at the current node and tries again in the next time step.

The next step is to choose the acceptance probability $\gamma_{ij} \in (0,1]$ such that

$$\sum_{i \in \mathcal{N}} \pi(i)\tilde{P}_{ij} = \sum_{i \in \mathcal{N}} \pi(i) Q_{ij} \gamma_{ij} = \kappa \pi(j) \tag{6}$$

is satisfied for all $j \in \mathcal{N}$, i.e., $\boldsymbol{\pi}\tilde{\mathbf{P}} = \kappa \boldsymbol{\pi}$ for some $\kappa > 0$. Then, by the Perron-Frobenius theorem, the distribution $\boldsymbol{\pi}$ is the *unique* left leading eigenvector of $\tilde{\mathbf{P}}$ and thus coincides with the unique QSD $\boldsymbol{\nu}$ of the transient chain $\mathbf{P}$, corresponding to the largest eigenvalue $\lambda = \kappa$. The key aspects here are (i) how to choose γ_{ij} using only locally available information and (ii) how to efficiently estimate or approximate the resulting QSD, or to obtain samples from the QSD in a distributed manner. In what follows, we present the specific form of γ_{ij} that maps any desired $\boldsymbol{\pi}$ on a given directed graph to the QSD $\boldsymbol{\nu}$ of the suitably constructed transient chain $\mathbf{P}$. The case of (ii) using crawling walks on a directed graph will be discussed later in Section 4.

For a given (proposed) ergodic, nonreversible chain $\mathbf{Q}$, we define $S_j := \{k \in \mathcal{N} \mid Q_{kj} > 0\}$ for each $j \in \mathcal{N}$. This is the set of states (nodes) from which the proposed chain $\mathbf{Q}$ can enter into state j in one step with positive probability. For each S_j, consider a probability distribution $\{\alpha_{kj}\}_{k \in S_j}$ such that $\alpha_{kj} > 0$ for all $k \in S_j$ and $\sum_{k \in S_j} \alpha_{kj} = 1$. Note that the set S_j is the support of the probability distribution $\{\alpha_{\cdot j}\}$. We then have the following.

Theorem 3.1. *For any choice of $\{\alpha_{kj}\}_{k \in S_j}$ on each S_j, $j \in \mathcal{N}$, if we set*

$$\gamma_{ij} \propto \frac{\pi(j)}{\pi(i)} \cdot \frac{\alpha_{ij}}{Q_{ij}}, \tag{7}$$

such that $\gamma_{ij} \leq 1$ for any pair (i,j) with $Q_{ij} > 0$, then the target distribution $\boldsymbol{\pi}$ becomes the QSD $\boldsymbol{\nu}$ of the transient chain $\mathbf{P}$, where $P_{ij} = Q_{ij}\gamma_{ij}$.

Proof. Let $\boldsymbol{\pi}$ be the given target distribution on $\mathcal{N}$. Fix $j \in \mathcal{N}$. Choose any arbitrary probability distribution $\{\alpha_{kj}\}_{k \in S_j}$ on the support S_j. By definition of S_j, we observe that for any $i \notin S_j$, i.e., $Q_{ij} = 0$, the choice of γ_{ij} is irrelevant since the proposed move from i to j will never take place. We can rewrite γ_{ij} in (7) as $\gamma_{ij} = \frac{1}{c} \cdot \frac{\pi(j)\alpha_{ij}}{\pi(i)Q_{ij}}$ with some suitable scaling constant $c > 1$ such that $\gamma_{ij} \leq 1$. Now, observe that

$$\sum_{i \in \mathcal{N}} \pi(i) P_{ij} = \sum_{i \in S_j} \pi(i) Q_{ij} \gamma_{ij} = \sum_{i \in S_j} \frac{1}{c} \pi(j) \alpha_{ij} = \frac{1}{c} \pi(j), \tag{8}$$

where the last equality is from $\sum_{k \in S_j} \alpha_{kj} = 1$. Since this holds for all $j \in \mathcal{N}$ and $\tilde{P}_{ij} = P_{ij}$ for $i, j \in \mathcal{N}$, we have $\boldsymbol{\pi}\tilde{\mathbf{P}} = (1/c)\boldsymbol{\pi}$ with $1/c > 0$. In view of the Perron-Frobenius theorem, $\boldsymbol{\pi}$ is the only positive eigenvector of $\tilde{\mathbf{P}}$ and is the leading left eigenvector, corresponding to the maximal eigenvalue $1/c$.

On the other hand, as explained in Section 3.1, from the Perron-Frobenius theorem we have the existence of a unique QSD $\boldsymbol{\nu}$ of the transient chain $\mathbf{P}$ that verifies $\boldsymbol{\nu}\tilde{\mathbf{P}} = \lambda \boldsymbol{\nu}$, with λ in (5). Therefore, the unique QSD $\boldsymbol{\nu}$ must coincide with our desired probability distribution $\boldsymbol{\pi}$, with $\lambda = 1/c \in (0,1)$. □

As a special case, we can set $\alpha_{kj} = 1/|S_j| = [\sum_{k \in \mathcal{N}} \mathbf{1}_{\{Q_{kj} > 0\}}]^{-1}$ for all $k \in S_j$, and then have the following expression for γ_{ij}.

Corollary 3.2. *Consider*

$$\gamma_{ij} \propto \frac{\pi(j)}{\pi(i)} \cdot \left[Q_{ij}|S_j|\right]^{-1} =: b_{ij}, \tag{9}$$

such that $\gamma_{ij} \leq 1$ for each (i,j) with $Q_{ij} > 0$. Then, the QSD $\boldsymbol{\nu}$ of $\mathbf{P}$ becomes identical to the target distribution $\boldsymbol{\pi}$.

While the probability distribution $\{\alpha_{\cdot j}\}$ on the support S_j can be made arbitrary as shown in Theorem 3.1, we consider γ_{ij} in (9) throughout the rest of the paper for its simplicity. Note that the acceptance probability γ_{ij} in (9) requires $\pi(j)/\pi(i)$, which is essentially the desired target distribution up to multiplicative constants, the proposal probability Q_{ij} (clearly known), and $|S_j|$ (the number of possible transitions into node j). These are all local information around i, j when the chain attempts to make a move from i to j.

As a special case, we consider $\mathbf{Q} = \mathbf{Q}^{\text{srw}}$, the transition matrix of the simple random walk on a given directed graph $\mathcal{G}$. In this case, $Q_{ij} = A_{ij}/d_i^+$ and $|S_j| = d_j^-$, the in-degree of node j. This is the only required local information, not exact knowledge (identities) of the nodes whose outgoing edges point to each node. Such in-degree can be obtained without much difficulty in online social networks having follower-followee relationship such as Google++, Twitter, and Instagram, since the in-degree is usually available as part of a user profile, which is simply the number of followers [73]. It is worth noting that the social networks are often much more stringent on retrieving the IDs of inlinks and outlinks of each user than looking up user profiles [73]. We shall even demonstrate in Section 6.3 that an online estimation of the in-degree of each node would suffice for our proposed NMMC method, even if such in-degree information may not be available.

Thanks to Theorem 3.1 and Corollary 3.2, without knowing the stationary distribution $\boldsymbol{\mu}$ of the simple random walk, which is *unknown* in general, we can achieve any desired distribution $\boldsymbol{\pi}$ by estimating the QSD of the transient chain $\mathbf{P} = [P_{ij}]$ on $\mathcal{N}^+$ with $P_{ij} = \gamma_{ij} Q_{ij} = \gamma_{ij} A_{ij}/d_i^+$, where

$$\gamma_{ij} \propto b_{ij} = \frac{\pi(j)}{\pi(i)} \cdot \frac{d_i^+}{d_j^-}, \tag{10}$$

with some scaling constant c such that $b_{ij}/c \leq 1$ for all $(i,j) \in \mathcal{E}$. Then, if we want to achieve the uniform distribution $\boldsymbol{\pi}$ on $\mathcal{G}$, i.e., $\pi(i) = 1/n$ for all $i \in \mathcal{N}$, then $c \geq \max_{(i,j)\in\mathcal{E}} d_i^+/d_j^-$. Since $d_j^- \geq 1$ for any $j \in \mathcal{N}$ (since the graph is irreducible), one easy choice would be $c = \max_{i\in\mathcal{N}} d_i^+$, i.e., the maximum out-degree of the graph. When such a global constant is known in advance, our Theorem 3.1, or Corollary 3.2, applies as is. In case that such constant is not available, later in Section 4.2, we will demonstrate that a real-time (dynamic) estimation of the global constant can also be safely used.

3.3 Mapping EVC to QSD

In addition to achieving any desired distribution $\boldsymbol{\pi}$ as a QSD, we demonstrate that we can go one step further to estimate the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$ of a given directed graph $\mathcal{G}$ in a similar fashion. As explained in Section 2.2, by definition, the EVC $\mathbf{x}$ is the (normalized) left leading eigenvector of the adjacency matrix $\mathbf{A}$ with the corresponding maximal eigenvalue $\lambda(A)$. Then, by simply adjusting γ_{ij} using local information, we can equate the resulting QSD with the EVC $\mathbf{x}$ of $\mathbf{A}$, as shown next.

Corollary 3.3. *If we set $\gamma_{ij} = A_{ij}/(cQ_{ij})$ for any (i,j) pair with $Q_{ij} > 0$, where c is some suitable scaling constant such that $\gamma_{ij} \leq 1$, then the QSD $\boldsymbol{\nu}$ of the transient $\mathbf{P}$ with elements $P_{ij} = Q_{ij}\gamma_{ij} = A_{ij}/c$ for $i, j \in \mathcal{N}$, becomes the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$.*

Proof. Using $\gamma_{ij} = A_{ij}/(cQ_{ij})$, and from (4) with P_{ij} replaced by $\tilde{P}_{ij}$, we observe that

$$\sum_{i \in \mathcal{N}} \nu(i) A_{ij} = (\lambda c) \cdot \nu(j) \iff \boldsymbol{\nu}\mathbf{A} = (\lambda c)\boldsymbol{\nu}, \tag{11}$$

with λ given in (5), i.e., the resulting QSD $\boldsymbol{\nu}$ is the left eigenvector of the adjacency matrix $\mathbf{A}$ with $\nu(i) > 0$, corresponding to the positive eigenvalue λc. The assertion then follows from the Perron-Frobenius theorem. □

Corollary 3.3 says that the well known EVC $\mathbf{x}$ of the adjacency matrix A of a given directed graph $\mathcal{G}$ can now be mapped to the unique QSD $\boldsymbol{\nu}$ of a transient Markov chain $\mathbf{P}$ driven by a general nonreversible chain $\mathbf{Q}$ using only local information for γ_{ij}, and that the corresponding largest eigenvalue $\lambda(\mathbf{A}) = \lambda c$ becomes the inverse of the epidemic threshold of the given graph (See Section 2.2.) For example, when the simple random walk is used, i.e., $Q_{ij} = Q_{ij}^{\mathsf{srw}} = A_{ij}/d_i^+$, we see that $\gamma_{ij} = d_i^+/c$, where $c = \max_{i \in \mathcal{N}} d_i^+$ can be set as the maximum out-degree of the graph.

So far, we have shown how to map any desired distribution π and the EVC $\mathbf{x}$ on a directed graph onto the QSD of some transient Markov chain $\mathbf{P}$, which is obtained from any ergodic, nonreversible chain $\mathbf{Q}$, with the simple random walk as an example. Now, the question is how to efficiently simulate or estimate the given QSD, or to obtain samples from the given QSD, via the same underlying nonreversible chain $\mathbf{Q}$ used for graph exploration. In the next section, we present several methods to simulate or estimate the QSD and complete our NMMC method using a set of non-Markovian random walks (or crawling agents) to accomplish the goal.

4 NMMC: QSD ESTIMATION

4.1 Non-Markovian Random Walks to Approximate QSD

We are often interested in calculating the expectation of a function $f : \mathcal{N} \to \mathbb{R}$ defined on the state space $\mathcal{N}$ with respect to a desired distribution $\boldsymbol{\pi}$, i.e., $\mathbb{E}_{\boldsymbol{\pi}}\{f\} = \sum_{i \in \mathcal{N}} f(i)\pi(i)$, by constructing an ergodic Markov chain $\{X_t\}$ that has the desired $\boldsymbol{\pi}$ as its equilibrium distribution. This is one of the main applications of the MCMC methods. By the ergodic theorem, we have

$$\lim_{t \to \infty} \frac{1}{t} \sum_{k=1}^{t} f(X_k) = \mathbb{E}_{\boldsymbol{\pi}}\{f\} \quad \text{a.s.}, \tag{12}$$

for any initial distribution for X_0. Thus, one can directly utilize this defining relationship by simulating the ergodic chain $\{X_t\}$ for a long time and taking the time average in order to estimate or approximate $\mathbb{E}_{\boldsymbol{\pi}}\{f\}$.

In case of the QSD $\boldsymbol{\nu}$ of a transient Markov chain $\{X_t\}$ on $\mathcal{N}^+ = \mathcal{N} \cup \{0\}$, we would also consider the long-time averages to estimate $\mathbb{E}_{\boldsymbol{\pi}}\{f\}$ driven by the desired QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ (or the QSD $\boldsymbol{\pi}$ itself) in view of

$$\lim_{t \to \infty} \mathbb{E}\left\{f(X_t) \mid X_0 \sim \boldsymbol{\mu}_0, X_t \neq 0\right\} = \mathbb{E}_{\boldsymbol{\pi}}\{f\}, \tag{13}$$

which is from (3) and the identity between QSD and QLD in the limit for any initial distribution $\boldsymbol{\mu}_0$. Unfortunately, however, this does not help much. In order to rely on the regularity inherent in the limiting relationship in (13), we would have to generate many trajectories for $\{X_t\}$ for a long time, only to find out that most of them get absorbed or trapped to the absorbing site '0' since $\mathbb{P}\{X_t \neq 0\}$ goes to zero exponentially fast, and we are left with very few trajectories that survive for a long time and possibly contribute to useful samples for the estimation. In short, this brute-force approach simply would not work for numerical estimation of the QSD in general.

In the mathematics literature, one practical approach in approximating the QSD is based on a historical urn-based process or a reinforced random walk on the state space. Although such a process loses the Markovian property due to the dependency on its own history or trajectory for the next move, using the past trajectory of the process is amenable to practical implementation. First, observe that from (4) and (5), the QSD $\boldsymbol{\nu}$ of the transient chain $\mathbf{P}$ satisfies

$$\nu(j) = \sum_{i\in\mathcal{N}} \nu(i)\left[P_{ij} + P_{i0}\nu(j)\right], \quad j \in \mathcal{N}. \tag{14}$$

This can be interpreted as an ergodic Markov chain on the state space $\mathcal{N}$ with its transition matrix $\mathbf{P}^{\boldsymbol{\nu}} = [P^{\boldsymbol{\nu}}_{ij}]$ given by

$$P^{\boldsymbol{\nu}}_{ij} := P_{ij} + P_{i0}\nu(j) = \tilde{P}_{ij} + P_{i0}\nu(j), \quad i,j \in \mathcal{N}. \tag{15}$$

That is to say, assuming that $\boldsymbol{\nu}$ is known, the process evolves according to $P_{ij} = \tilde{P}_{ij}$ until it gets absorbed, upon which it instantaneously 'redistributes' to a new node in $\mathcal{N}$ according to $\boldsymbol{\nu}$. Let $\Phi(\boldsymbol{\nu})$ be the invariant (stationary) distribution of this ergodic Markov chain with its transition matrix $\mathbf{P}^{\boldsymbol{\nu}} = [P^{\boldsymbol{\nu}}_{ij}]$, indexed with $\boldsymbol{\nu}$. Then, we can see that the QSD $\boldsymbol{\nu}$ in (14) becomes precisely the fixed point of the mapping from $\boldsymbol{\nu}$ to $\Phi(\boldsymbol{\nu})$. In other words, $\boldsymbol{\nu}$ is the QSD for $\mathbf{P}$ if and only if $\Phi(\boldsymbol{\nu}) = \boldsymbol{\nu}$. The construction of such an ergodic chain is, however, practically infeasible in estimating the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$, since lack of the entire topological information prohibits direct access into the state space $\mathcal{N}$ according to $\boldsymbol{\pi}$, or $\boldsymbol{\pi}$ is simply unknown, e.g., for EVC of $\mathbf{A}$.

The above observation nonetheless motivated one of the earliest efforts in approximating the QSD of a given transient chain by Aldous *et al.* [1] to replace $\boldsymbol{\nu}$ (unknown) in (14) with the (known) empirical distribution of the history of the original process.[4] To be precise, we define its historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ by time t of the underlying stochastic process $\{Z_t\}_{t\geq 0}$ as

$$\hat{\boldsymbol{\mu}}_t = \frac{1}{t+1}\sum_{k=0}^{t} \delta_{Z_k}, \tag{16}$$

with $\hat{\boldsymbol{\mu}}_t = [\hat{\mu}_t(1), \cdots, \hat{\mu}_t(n)]$, where $\hat{\mu}_t(i)$ is simply given by

$$\hat{\mu}_t(i) = \frac{1}{t+1}\sum_{k=0}^{t} \delta_{Z_k}(i) = \frac{1}{t+1}\sum_{k=0}^{t} \mathbf{1}_{\{Z_k=i\}},$$

which records the relative visit frequency of $\{Z_k\}$ to the state $i \in \mathcal{N}$ over the first $t+1$ time steps. Note that $\hat{\mu}_t(i)$ is random while $\sum_{i\in\mathcal{N}} \hat{\mu}_t(i) = 1$ for each t, i.e., $\hat{\boldsymbol{\mu}}_t$ is in fact a probability distribution (vector) on $\mathcal{N}$ for each t.

Consider the natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$ associated with the historical process $\{Z_t\}$, i.e., $\mathcal{F}_t := \sigma\{Z_k \mid k \leq t\}$. We then have

$$\mathbb{P}\{Z_{t+1} = j \mid \mathcal{F}_t\} = \sum_{i\in\mathcal{N}} P^{\hat{\boldsymbol{\mu}}_t}_{ij} \cdot \mathbf{1}_{\{Z_t=i\}}$$

for any given $j \in \mathcal{N}$, where $P^{\hat{\boldsymbol{\mu}}_t}_{ij}$ is from (15) with $\boldsymbol{\nu}$ replaced by $\hat{\boldsymbol{\mu}}_t$. That is to say, given the full history $\mathcal{F}_t$ of the process $\{Z_t\}$ up to time t, the process will choose its next state Z_{t+1} by the original transition matrix $\mathbf{P}$ if not getting absorbed. When it hits the absorbing state '0', it will

[4] In the literature, there are some study on Fleming-Viot (FV) processes and the evolution of the associated spatial empirical measures converging to the QSD [3, 33–36, 49]. It is essentially to replace the unknown $\boldsymbol{\nu}$ in (14) by the 'spatial' occupancy measure of a large number of particles undergoing identical dynamics with mutual interaction upon particle absorption. To be more precise, in our graph setting, it can be described as follows. There are K particles or agents, each of which moves according to $\mathbf{P}$ of the transient Markov chain on $\mathcal{N}^+$. When any of the particles gets absorbed by '0', it is instantaneously relocated to one of the remaining particles, chosen uniformly at random, on $\mathcal{N}$. Then, the empirical measure of the positions of the particles converges to the QSD as *both* the number of particles K and time t go to infinity. The number of particles required under the FV method, however, should be much larger than the size of the graph (i.e., $K \gg n$), making the FV method impractical to our scenario. Thus, we do not consider the FV method in this paper.

instantaneously jump to another state in $\mathcal{N}$ with probability determined by the historical empirical distribution $\hat{\boldsymbol{\mu}}_t = \frac{1}{t+1}\sum_{k=0}^{t}\delta_{Z_k}$, observed by the current time instant t and thus readily available to us.

This method has then been recently generalized in [10, 11] by utilizing the stochastic approximation theory and its induced differential equations developed for reinforced random walks [9, 55], allowing different forms of the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ that substitutes $\boldsymbol{\nu}$ in (15). Specifically, the distribution $\hat{\boldsymbol{\mu}}_t$ is generalized as

$$\hat{\boldsymbol{\mu}}_t = \frac{\sum_{k=0}^{t} w_k \delta_{Z_k}}{\sum_{k=0}^{t} w_k} = \sum_{k=0}^{t} \eta_t(k)\delta_{Z_k}, \tag{17}$$

with

$$\eta_t(k) := \frac{w_k}{\sum_{k'=0}^{t} w_{k'}}, \qquad k = 0, 1, 2, \ldots, t. \tag{18}$$

Here, $\hat{\mu}_t(i)$ indicates the visit frequency to node i by $\{Z_k\}$ up to time t, each of which is *weighted* by $\eta_t(k)$. This makes the redistribution dynamics of the process $\{Z_t\}$ different, while the other remains the same as above. When the process hits the absorbing state at time $t+1$, it relocates directly to one of the previously-visited nodes according to $\hat{\boldsymbol{\mu}}_t$ in (17). This can also be interpreted as choosing $Z_{t+1} := Z_k$ with probability $\eta_t(k)$ in (18), $k = 0, 1, \ldots, t$. Setting $w_k = 1$, or $\eta_t(k) = 1/(t+1)$, for all k in (17) reduces to (16).

The resulting process $\{Z_t\}$ is clearly non-Markovian and history-dependent. It is shown in [10, 11] that for a large class of non-decreasing weight functions w_k, or sequences $\{\eta_t(t)\}_{t\geq 0}$,

$$\lim_{t\to\infty} \hat{\boldsymbol{\mu}}_t = \boldsymbol{\nu} \ \text{ a.s., and } \ \lim_{t\to\infty} \mathbb{E}\{\hat{\boldsymbol{\mu}}_t\} = \boldsymbol{\nu}, \tag{19}$$

for any arbitrary initial state Z_0 with $\hat{\boldsymbol{\mu}}_0 = \delta_{Z_0}$. Note that the allowed weight function w_k includes, for example, $w_k = 1$ (constant), $w_k = k^a$ (polynomial), and 2^{k^b} (sub-exponential), where $a > 1$ and $b \in (0, 1)$. These weight functions make the sequence $\{\eta_t(t)\}_{t\geq 0}$ non-increasing in t and satisfy the required conditions in [10, 11]. We adopt this history-based process with reinforcement to estimate the desired QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ as an integral part of our NMMC method.

The basic operation of our NMMC method to achieve any desired distribution $\boldsymbol{\pi}$ on a directed graph $\mathcal{G}$ or to estimate the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$ can be summarized as follows. Under our NMMC method, a set of non-Markovian random-walk agents are launched to crawl the graph, and each of them independently moves over $\mathcal{G}$ by choosing the next node Z_{t+1} at the current node $Z_t = i$ according to the transition probability $P_{ij} = Q_{ij}\gamma_{ij}$, before it gets absorbed and relocated. In other words, a move from node i to j is *proposed* with probability Q_{ij} following a directed edge (i, j) on $\mathcal{G}$. This move is then *accepted* with probability γ_{ij} as given in Theorem 3.1 and Corollary 3.2 for achieving $\boldsymbol{\pi}$ (or Corollary 3.3 for estimating the EVC $\mathbf{x}$). Otherwise, it is *rejected* in which case the agent is instantaneously relocated (or redistributed) to one of the previously-visited nodes according to the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ in (17). Figure 1 illustrates the overall operation of our NMMC method.

We note that the history-dependent/reinforced random walks are quite relevant and amenable to practical implementation on most popular online social networks, since the crawler (e.g., a simple random walk) continues to collect the ID (e.g., URL) of each node visited as it explores the graph and can always relocate/redistribute to one of the previously-visited nodes, while keeping track of which nodes have been visited up until now. The historical empirical distribution governs not only the dynamics of each crawling agent, but it also enables us to estimate the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$. Therefore, our NMMC method provides the same benefit and fruitful denouement that we cherished from the standard MCMC methods that are rooted in the ergodic theorem in (12).

Algorithm 1 Non-Markovian Monte Carlo (at time $t \geq 1$)

```
Input: $\mathbf{Q}$, $\boldsymbol{\nu}$, $\hat{\boldsymbol{\mu}}_0$, and $w_k$
i ← $Z_{t-1}$          // the node at time $t-1$
Select node $j$ with probability $Q_{ij}$
Generate $u \sim U(0,1)$
if $u \leq \gamma_{ij}$ then
    $Z_t \leftarrow j$          // the proposed move is accepted
else
    /* the agent is relocated according to $\hat{\boldsymbol{\mu}}_t$ */
    Select node $k$ with probability $\hat{\mu}_t(k)$
    $Z_t \leftarrow k$
end if
/* update the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ */
$\hat{\boldsymbol{\mu}}_t \leftarrow [1-\eta_t(t)]\,\hat{\boldsymbol{\mu}}_{t-1} + \eta_t(t)\delta_{Z_t}$
```

4.2 Performance Enhancements

So far, we have demonstrated how any ergodic (inherently nonreversible) Markov chain $\mathbf{Q}$ with unknown $\boldsymbol{\mu}$ can be used to achieve a given desired distribution $\boldsymbol{\pi}$ on a directed graph $\mathcal{G}$ and to estimate the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$. It is essentially achieved by combining our mapping $\boldsymbol{\pi}$ or $\mathbf{x}$ to the QSD $\boldsymbol{\nu}$ of a related transient Markov chain and the recent development in the mathematics literature as to estimating the QSD via a set of non-Markov chain, history-dependent random walks.

As shall be seen later in Section 5, the direct application of the non-Markovian random walks with historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ may not be satisfactory due to a possible imbalance between their diffusion and redistribution, both of which are the necessary integral components. In this section, we present several ways to make the crawling walks achieve a right balance between their diffusion and redistribution, which translates into speeding up the convergence of $\hat{\boldsymbol{\mu}}_t$ to the desired QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ (or $\boldsymbol{\nu} = \mathbf{x}$).

Initial empirical measure $\hat{\boldsymbol{\mu}}_0$. We next revisit the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ in (17). Letting $W_t := \sum_{k=0}^{t} w_k$ for a given positive weight sequence, $\{w_k\}_{k\geq 0}$, we observe that (17) can be written as, for $t \geq 1$,

$$W_t \hat{\boldsymbol{\mu}}_t = W_{t-1}\hat{\boldsymbol{\mu}}_{t-1} + w_t \delta_{Z_t}, \tag{20}$$

which then leads to

$$\begin{aligned} \hat{\boldsymbol{\mu}}_t &= \frac{W_{t-1}}{W_t}\hat{\boldsymbol{\mu}}_{t-1} + \frac{w_t}{W_t}\delta_{Z_t} = \left(1 - \frac{w_t}{W_t}\right)\hat{\boldsymbol{\mu}}_{t-1} + \frac{w_t}{W_t}\delta_{Z_t} \\ &= [1-\eta_t(t)]\,\hat{\boldsymbol{\mu}}_{t-1} + \eta_t(t)\delta_{Z_t}, \end{aligned} \tag{21}$$

from the identities $W_t = W_{t-1} + w_t$ and $\eta_t(t) = w_t/W_t$. This implies that the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ can be computed recursively based on (21). Although the convergence of $\hat{\boldsymbol{\mu}}_t$ to the QSD $\boldsymbol{\nu}$ was shown for any arbitrary initial state Z_0 with $\hat{\boldsymbol{\mu}}_0 = \delta_{Z_0}$ in [10, 11], its proof based on the stochastic approximation theory and the induced differential equations in fact does not require the initial distribution $\hat{\boldsymbol{\mu}}_0$ in (21) to be in the form of δ_{Z_0}, where $\hat{\boldsymbol{\mu}}_0$ has a value of 1 at the element Z_0 and zero at all other elements, but can be made arbitrary as long as $\hat{\boldsymbol{\mu}}_0$ is a probability distribution (vector) on $\mathcal{N}$.

This freedom on the choice of $\hat{\boldsymbol{\mu}}_0$ gives us a great degree of flexibility. Clearly, the choice of $\hat{\boldsymbol{\mu}}_0$ affects the whole evolution of the process $\{Z_t\}$ for all time t via redistribution mechanism,

Algorithm 2 Dynamic Non-Markovian Monte Carlo

```
/* This pseudocode only replaces lines 3–10 of Algorithm 1 */
Generate $u_1, u_2 \sim U(0,1)$
if $u_1 \leq p$ and $c_{t-1} < b_{ij}$ then
    $c_t \leftarrow b_{ij}$        // $c_t$ is updated by $b_{ij}$
else
    $c_t \leftarrow c_{t-1}$        // $c_{t-1}$ is kept
end if
if $u_2 \leq \min\{1, b_{ij}/c_t\}$ then
    $Z_t \leftarrow j$        // the proposed move is accepted
else
    Select node $k$ with probability $\hat{\mu}_t(k)$
    $Z_t \leftarrow k$        // the agent is redistributed
end if
```

and thus can be made to our advantage. For example, a judicious choice of $\hat{\boldsymbol{\mu}}_0$ can translate into a situation where a crawling agent $\{Z_t\}$ can start with an 'already-established' history to some degree or more generally, with any *arbitrary (normalized) visit counts* assigned to a subset of nodes on the graph to begin with. Thus, our proposed NMMC method can be generalized with arbitrary $\hat{\boldsymbol{\mu}}_0$ and its overall procedure is now summarized in Algorithm 1. Here the proposed chain $\mathbf{Q}$, the initial empirical measure $\hat{\boldsymbol{\mu}}_0$, and the (unnormalized) weight function w_k should be specified as inputs into Algorithm 1. Note that the initial position of a crawling agent Z_0 is randomly chosen according to $\hat{\boldsymbol{\mu}}_0$.

Dynamic and adaptive choice of parameters. Our proposed NMMC method is mathematically guaranteed but its speed of convergence depends on the form of the acceptance probability γ_{ij}, along with Q_{ij} and the target QSD, which interweaves diffusion and redistribution – two seemingly opposite (yet integral) components of the method. As we have seen from Section 3, the specific form of γ_{ij} is different depending on the choice of $\mathbf{Q} = [Q_{ij}]$ and what to estimate via the target QSD, i.e., the target distribution $\boldsymbol{\pi}$ or the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$. It is particularly desirable to maintain some reasonable values of γ_{ij} over almost every $(i,j) \in \mathcal{E}$, or more precisely, transition pair (i,j) with $Q_{ij} > 0$, so as to strike the right balance between diffusion and redistribution to achieve faster convergence.

We take a closer look at the acceptance probability γ_{ij} as a function of Q_{ij} and the target QSD. Recall that from Theorem 3.1 and Corollary 3.2, it is given by $\gamma_{ij} = b_{ij}/c$ in (9) for achieving (or sampling from) a given distribution $\boldsymbol{\pi}$ on the directed graph $\mathcal{G}$, and the corresponding normalizing constant c can be obtained based on $c = \max_{i,j} b_{ij}$ (the smallest possible value for c). Similarly, for estimating the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$, it can be chosen as $c = \max_{i,j} 1/Q_{ij}$ for $\gamma_{ij} = A_{ij}/(cQ_{ij})$ from Corollary 3.2. The global constant c over the graph, however, may not be available in advance or could take a long time to estimate, although its real-time (dynamic) estimation can be incorporated as shall be explained shortly. Furthermore, the resulting values of γ_{ij} may vary over $(i,j) \in \mathcal{E}$ and they can also be small for some (i,j) pairs. Such small acceptance probabilities in turn make the agent get absorbed to '0' and redistributed to the already visited nodes. The redistribution mechanism is of course an important and necessary component of our algorithm to shape up the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ toward the target QSD. But, if it is too strong (very small γ_{ij}), the agent may virtually get stuck on a small set of visited nodes before diffusing out to a potentially vast unvisited portion of the graph (state space) for exploration.

To remedy this possibly slow diffusion problem, we observe that the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ when achieving any desired distribution $\boldsymbol{\pi}$ is completely controllable to our benefit. To be precise, if it is of primary concern to estimate $\mathbb{E}_{\boldsymbol{\pi}}\{f\} = \sum_{i \in \mathcal{N}} f(i)\pi(i)$ for a given sampling function $f : \mathcal{N} \to \mathbb{R}$, which is often the case for 'undirected' graph sampling [5, 32, 37, 44, 58, 65], the target distribution can be *controlled* together with a so-called importance sampling method in MCMC [47]. With an alternative sampling distribution $\boldsymbol{\pi}'$, it is to estimate

$$\mathbb{E}_{\boldsymbol{\pi}'}\left\{f\frac{\boldsymbol{\pi}}{\boldsymbol{\pi}'}\right\} = \sum_{i \in \mathcal{N}} f(i)\frac{\pi(i)}{\pi'(i)}\pi'(i) = \mathbb{E}_{\boldsymbol{\pi}}\{f\}. \tag{22}$$

Thus, we can choose the target distribution $\boldsymbol{\pi}$ along with $\mathbf{Q}$ so that it translates into more balanced γ_{ij}. As shall be demonstrated later in Section 5, with $\mathbf{Q} = \mathbf{Q}^{\text{srw}}$, setting the target $\boldsymbol{\pi}$ to be proportional to node in-degrees, i.e., $\pi(i) \propto d_i^-, i \in \mathcal{N}$, turns out to be very effective in achieving fast convergence speed. This improvement is somewhat expected since the simple random walk would visit high in-degree nodes more frequently, which helps the historical empirical distribution $\hat{\boldsymbol{\mu}}$ get closer to this target $\boldsymbol{\pi}$ quickly.

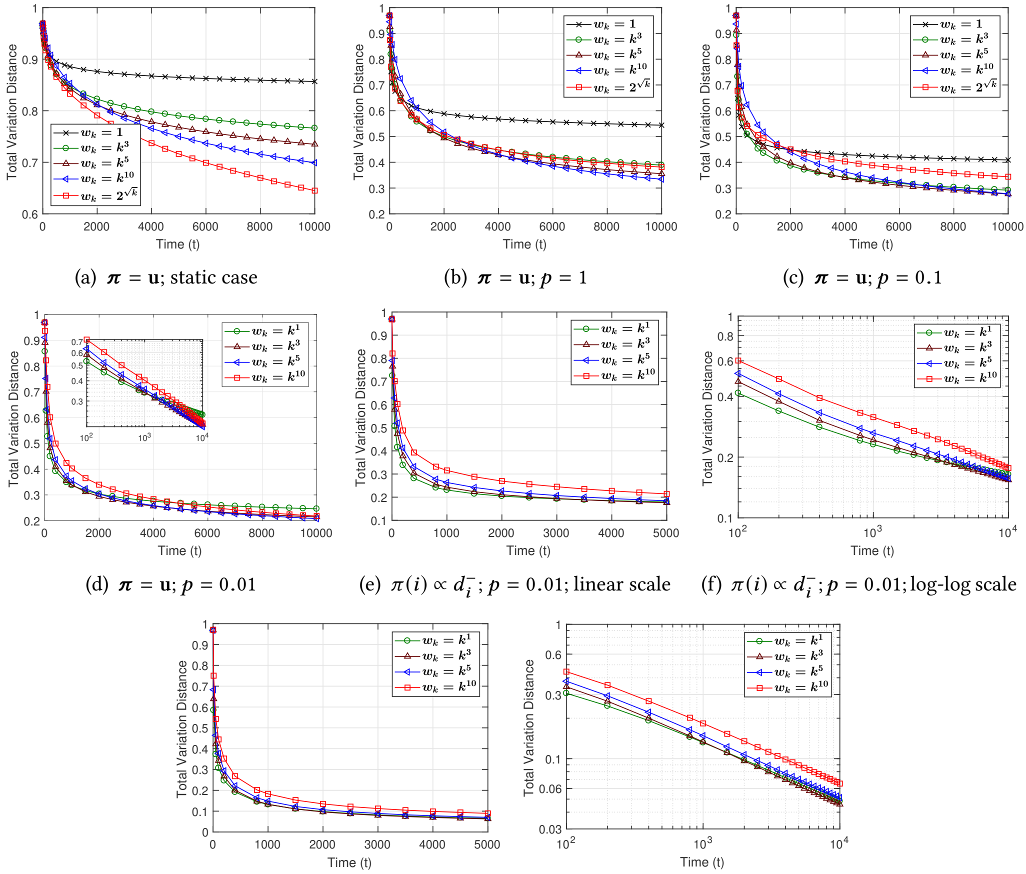
To further alleviate the slow diffusion problem and also enable online estimation of the normalizing constant c, we allow c to be time-varying, say c_t, so that it can be learned through graph exploration by the crawling agent. For ease of exposition, we focus on achieving the desired $\boldsymbol{\pi}$ over the graph, but everything can easily be carried over to estimating the EVC $\mathbf{x}$ of $\mathbf{A}$. We set the acceptance probability $\gamma_{ij}(t) := b_{ij}/c_t$, which is updated in a way that $c_t := b_{ij}$ at every time t if $c_{t-1} < b_{ij}$ for a pair of the current node i and its neighbor j as a potential next node, and $c_t := c_{t-1}$ if otherwise. In other words, the agent keeps track of the maximum c_t over all the values of b_{ij} for node pairs that have been discovered up to time t. Note that $c_0 = 1$ initially.

We can take one step further to foster the initial diffusion speed to a greater extent, by intentionally slowing down the growth of c_t for higher values of $\gamma_{ij}(t)$ (being closer to one). Specifically, with probability $p > 0$, we perform the aforementioned update of c_t, while with probability $1-p$, the time-varying normalizing constant $c_t = c_{t-1}$ is kept the same even if $c_{t-1} < b_{ij}$, in which case the proposed move to j is always accepted. The entire procedure of 'dynamic' NMMC method is summarized in Algorithm 2, which replaces the lines 3–10 of Algorithm 1. Note that the resulting acceptance probability becomes $\gamma_{ij}(t) = \min\{1, b_{ij}/c_t\}$ as seen from Algorithm 2.

While the NMMC method with time-varying c_t is seemingly heuristic, it is in fact well supported by Theorem 3.1 and our observation on the initial empirical measure $\hat{\boldsymbol{\mu}}_0$, i.e., *the theoretical guarantee remains intact.* To see this, we define a stopping time as $t^* := \min\{t \geq 0 \mid c_t = c\}$. Since the graph is finite, this t^* is almost surely finite for $p > 0$. We would run the NMMC method with c_t until t^*, and then reset the time clock to zero at t^* and treat the historical empirical distribution $\hat{\boldsymbol{\mu}}_{t^*}$ to be newly set as $\hat{\boldsymbol{\mu}}_0$. It then follows from our observation pertaining to $\hat{\boldsymbol{\mu}}_0$, along with Theorem 3.1, that the historical empirical measure $\hat{\boldsymbol{\mu}}_t$ converges to the given target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$. In addition to the theoretical guarantees, we next demonstrate that our NMMC method with a suitable choice of the c_t-updating probability p indeed greatly improves the speed of convergence to the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ by effectively combating the slow-diffusion problem triggered by overly frequent redistribution to already-visited sites.

5 SIMULATION RESULTS

In this section, we present simulation results to support our theoretical findings. To this end, we consider four real-world network datasets available in the SNAP repository [46] as test-case directed graphs, whose statistics are summarized in Table 1. The numbers of nodes and edges in the largest strongly connected component (LSCC) of each graph are given in the fourth and fifth



(a) $\boldsymbol{\pi} = \mathbf{u}$; static case

(b) $\boldsymbol{\pi} = \mathbf{u}$; $p = 1$

(c) $\boldsymbol{\pi} = \mathbf{u}$; $p = 0.1$

(d) $\boldsymbol{\pi} = \mathbf{u}$; $p = 0.01$

(e) $\pi(i) \propto d_i^-$; $p = 0.01$; linear scale

(f) $\pi(i) \propto d_i^-$; $p = 0.01$; log-log scale

(g) EVC $\mathbf{x}$; $p = 0.01$; linear scale

(h) EVC $\mathbf{x}$; $p = 0.01$; log-log scale

Fig. 2. The TVD results obtained under the Gnutella graph.

columns respectively. Note that self-loops, if present, e.g., in the original dataset of Slashdot graph, are removed for our simulations.

	# nodes	# edges	# nodes (LSCC)	# edges (LSCC)
Gnutella	8,846	31,839	3,234	13,453
Slashdot	82,168	948,464	71,307	912,381
Wiki-Talk	2,394,385	5,021,410	111,881	1,477,893
Amazon	400,727	3,200,440	380,167	3,069,889

Table 1. Statistics of the datasets

The primary focus here is, through numerical simulations, to confirm that the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ by our NMMC method in Algorithms 1 and 2 (the original one and its dynamic extension) converges to the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ (or $\boldsymbol{\nu} = \mathbf{x}$) and to evaluate the speed of convergence of $\hat{\boldsymbol{\mu}}_t$ to $\boldsymbol{\nu}$. As a performance metric, we use the total variation distance (TVD)[5] between the historical

[5] Note that our purpose here is completely different from the typical use of TVD for characterizing the speed of convergence of an ergodic, reversible Markov chain on a finite state-space, say $\mathbf{Q}$, to its stationary distribution, say $\boldsymbol{\mu}$. To be precise, for any initial $\boldsymbol{\mu}_0$, $\|\boldsymbol{\mu}_0 \mathbf{Q}^t - \boldsymbol{\mu}\|_{TV} \leq \kappa \rho^t$ for some constant κ, where ρ is the second largest eigenvalue modulus of $\mathbf{Q}$ with

empirical distribution $\hat{\boldsymbol{\mu}}_t$ and the target QSD $\boldsymbol{\nu}$, which is given by

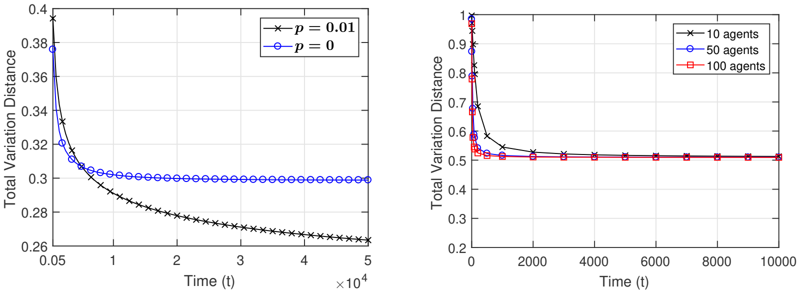
$$\|\hat{\boldsymbol{\mu}}_t - \boldsymbol{\nu}\|_{TV} \triangleq \max_{E \subseteq \mathcal{N}} |\hat{\mu}_t(E) - \nu(E)| = \frac{1}{2} \sum_{i \in \mathcal{N}} |\hat{\mu}_t(i) - \nu(i)|. \tag{23}$$

The inputs of Algorithms 1 and 2 are chosen as follows. The initial position Z_0 of each agent is randomly determined, with the initial empirical measure $\hat{\boldsymbol{\mu}}_0 = \delta_{Z_0}$. For the target QSD $\boldsymbol{\nu}$ to be mapped and estimated, we consider (i) uniform distribution, i.e., $\pi(j) = 1/n$, (ii) in-degree distribution, i.e., $\pi(j) \propto d_j^-$, and (iii) the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$. We also consider $\mathbf{Q} = \mathbf{Q}^{\text{srw}}$, where $Q_{ij} = A_{ij}/d_i^+$ for $i, j \in \mathcal{N}$. We use the LSCC of each graph for simulations of all these cases. In order to understand how different choices of unnormalized weight functions w_k would affect the speed of convergence, we vary weight functions as $w_k = 1, k^1, k^3, k^5, k^{10}, 2^{\sqrt{k}}$, with increasing tendency to relocate to recently-visited nodes upon redistribution. For each simulation we use a set of independent crawling agents, which are independently moving over the graph. While each agent maintains its own historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ for redistribution in case of individual absorption, all the historical empirical distributions are combined and normalized as a unified distribution for the convergence test.

We first present simulation results for Gnutella graph. We use 100 independent crawling agents for all the results. Figure 2(a) shows the TVD results when achieving a uniform distribution $\mathbf{u}$ under the NMMC method in Algorithm 1, where the global constant c is known a priori. Recall that for achieving $\boldsymbol{\pi} = \mathbf{u}$, the acceptance probability γ_{ij} in (9) becomes $\gamma_{ij} = b_{ij}/c$ with $b_{ij} = d_i^+/d_j^-$ and $c = \max_{i,j} b_{ij}$. This global constant c can be as large as $c = \max_i d_i^+$ and this was the case in this simulation. Figure 2(a) shows that $\hat{\boldsymbol{\mu}}_t$ is getting closer to the target QSD $\boldsymbol{\nu} = \boldsymbol{\pi}$ for all considered weight functions w_k. In addition, Figure 2(a) exhibits that heavier emphasis on latter visited sites leads to faster convergence of $\hat{\boldsymbol{\mu}}_t$ to $\boldsymbol{\pi}$. We would expect that it is attributed to higher weights of being relocated toward more recently-visited nodes that gives rise to faster diffusion on $\mathcal{G}$ and quicker exploration of the graph. Nonetheless, the overall performance may be considered unsatisfactory as the resulting TVD values are still high. This means that the redistributions still take place too often, resulting in an imbalance between diffusion and redistribution.

In Figure 2(b)–(d), we present the TVD results for estimating the same uniform $\mathbf{u}$ under the *dynamic* NMMC method in Algorithm 2 while varying the value of the c_t-updating probability p from 1 to 0.01. It turns out that, with the choice of $p = 0.01$, the TVD is reduced by more than 0.1 at every time step when compared to that of $p = 1$. To achieve the same precision in terms of the TVD, the number of time steps t required for $p = 1$ would be much larger than that of $p = 0.01$ by at least an order of magnitude. Observing a clear decreasing trend with p, one may think that $p = 0$ would be the best. However, when $p = 0$, c_t is never updated over t as seen from our Algorithm 2, i.e., $c_t = c_0 = 1$ for all t. Still, it is possible that $\gamma_{ij}(t) = \min\{1, b_{ij}/c_t\} = \min\{1, b_{ij}\}$ is less than one, meaning that a proposed move is not always accepted. Nonetheless, since c_t never approaches the true value of c when $p = 0$, there is no theoretical guarantee for the convergence either. Accordingly, the resulting TVD does not decrease as shown in Figure 3(a), where $w_k = 1$ is considered. Furthermore, in view of faster diffusion with smaller p leading to a better speed of convergence, we can take another extreme case by simply running the simple random walk all the time, which always diffuses without any redistribution mechanism. It turns out that the resulting TVD again does not decrease as shown in Figure 3(b). The observations so far collectively indicate

$\rho < 1$ [15]. This exponential convergence speed is for the 'marginal' distribution $\boldsymbol{\mu}_0\mathbf{Q}^t$ at time t to $\boldsymbol{\mu}$ as t increases. In contrast, we are interested in the convergence of the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$, which encompasses the entire history of the underlying process with proper time weights.



(a) $p = 0$ vs. $p = 0.01$

(b) $\mathbf{Q}^{\mathrm{srw}}$ with no redistribution

Fig. 3. Failure of convergence $\hat{\boldsymbol{\mu}}_t$ to $\boldsymbol{\pi} = \mathbf{u}$ when using two possible extreme methods that put too much weight on diffusion of the underlying crawling agent.

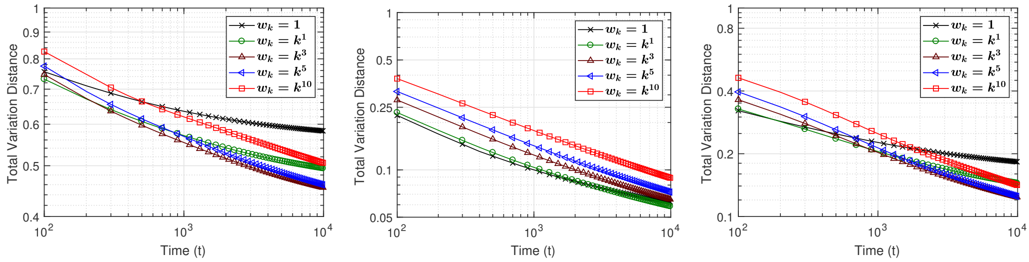
an importance of having a right balance between diffusion and redistribution to shape up the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ toward the target QSD and to achieve faster convergence.

We further take a closer look into the cases with $w_k = k^\alpha$ and $w_k = 2^{\sqrt{k}}$. While the case with $w_k = 2^{\sqrt{k}}$ is the best with no dynamic update of c as seen from Figure 2(a), it becomes worse than the case with $w_k = k^\alpha$ when using the dynamic NMMC method with c_t-updating probability p as shown in Figure 2(b)–(c). The case with $w_k = 2^{\sqrt{k}}$ when combined with the dynamic updates of c_t places over-emphasis on diffusion, which causes more harm than help. This again signifies the importance of striking the right balance between diffusion and redistribution of the reinforced walk Z_t toward faster convergence, although the presence of such a crossover would depend on the choice of the directed graph $\mathcal{G}$, target QSD $\boldsymbol{\nu}$ and the proposal chain $\mathbf{Q}$. Furthermore, we have observed that the case with $w_k = k^\alpha$ is always better with the use of c_t-updating probability p than the case with $w_k = 2^{\sqrt{k}}$, when achieving the in-degree distribution, i.e., $\pi(j) \propto d_j^-$, and the EVC $\mathbf{x}$ of the adjacency matrix $\mathbf{A}$. We omit the results for brevity and exclude the case with $w_k = 2^{\sqrt{k}}$ for the subsequent numerical evaluations. Note that the acceptance probability γ_{ij} in (9) becomes $\gamma_{ij} = d_i^+/(cd_i^-)$ with $c = \max_i d_i^+/d_i^-$ for the former, and $\gamma_{ij} = d_i^+/c$ with $c = \max_i d_i^+$ for the latter.

We instead report the results under the (unnormalized) weight function $w_k = k^\alpha$ with different values of α and $p = 0.01$ for all three cases of the target QSD in Figure 2(d)–(h). For achieving $\boldsymbol{\pi} = \mathbf{u}$, the case with $\alpha = 1$ exhibits the fastest decaying rate in its TVD curve initially, but later becomes slower than the other cases. Similar tendency with $\alpha = 1$ is observed for achieving $\pi(j) \propto d_j^-$ and estimating the EVC $\mathbf{x}$, as shown in Figure 2(e)-(h). On the other hand, we see that the case with $\alpha = 10$ eventually becomes the best with the fastest decaying rate, as we can see from the inset figure of Figure 2(d). We also observe that the eventual winner is the case with $\alpha = 5$ for achieving $\pi(j) \propto d_j^-$, while the case with $\alpha = 3$ eventually appears to be the best for estimating the EVC $\mathbf{x}$. This set of simulation results indicate that target QSD $\boldsymbol{\nu}$ clearly affects the extent of diffusion to achieve the right balance between diffusion and redistribution.

In addition, we observe that TVD values decay to zero polynomially fast, as evidenced in Figure 2(f), Figure 2(h), and the inset figure of Figure 2(d), all drawn on a log-log scale. We have also observed such a polynomial decaying behavior for the other choices of parameters. This behavior can be explained as follows.

Suppose that we have a trajectory of an ergodic Markov chain, say $\{X_t\}_{t \geq 0}$, with its stationary distribution $\boldsymbol{\pi}$. For a function $f : \mathcal{N} \to \mathbb{R}$, letting $S_t := \sum_{k=1}^{t} f(X_k)$, S_t/t converges to $\mathbb{E}_{\boldsymbol{\pi}}\{f\}$ almost surely as seen from (12) as t grows. In addition, the standard Central Limit Theorem (CLT)



(a) $\boldsymbol{\pi} = \mathbf{u}$; $p = 0.01$ (b) $\pi(i) \propto d_i^-$; $p = 0.01$ (c) EVC $\mathbf{x}$; $p = 0.01$

Fig. 4. The TVD results obtained under Slashdot graph.

for ergodic Markov chains says that for any bounded function f and any initial distribution on X_0,

$$\sqrt{t}\,(S_t/t - \mathbb{E}_{\boldsymbol{\pi}}\{f\}) \xrightarrow{d} N(0,\sigma^2), \text{ as } t \to \infty, \tag{24}$$

where $N(0,\sigma^2)$ is a Gaussian random variable with zero mean and variance σ^2. The (asymptotic) variance σ^2 is given by [5, 44, 47]

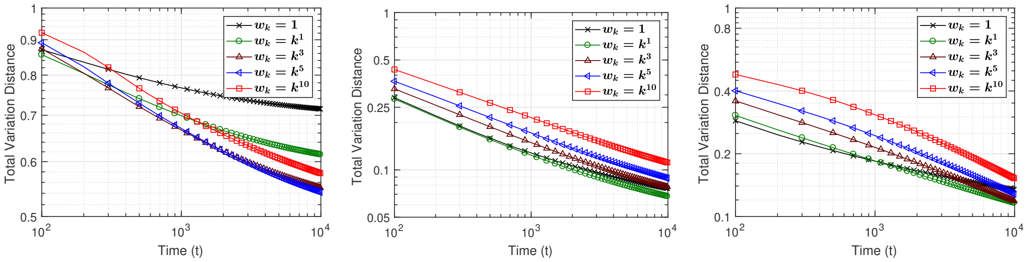
$$\sigma^2 = \lim_{t\to\infty} \frac{1}{t}\mathrm{Var}\,(S_t) = \mathrm{Var}_{\boldsymbol{\pi}}\{f\} + 2\sum_{k=1}^{\infty} \mathrm{Cov}_{\boldsymbol{\pi}}\,(f(X_0), f(X_k)),$$

where the first term in RHS is the marginal variance of f with respect to $\boldsymbol{\pi}$ and the second term in RHS is the lag-k autocovariance of the stationary sequence $\{f(X_t)\}$. From (24), we see that the fluctuations of S_t/t around the mean value $\mathbb{E}_{\boldsymbol{\pi}}\{f\}$ are roughly within the interval of length of order $\sigma/\sqrt{t}$. Even when X_t's are *i.i.d.* over t, each having distribution $\boldsymbol{\pi}$ (i.e., perfect independent samples all the time), the asymptotic order of convergence from S_t/t to $\mathbb{E}_{\pi}\{f\}$ still remains the same as $O(1/\sqrt{t})$, while σ^2 reduces to $\mathrm{Var}_{\boldsymbol{\pi}}\{f\}$.

What we have in this paper is the convergence of $\hat{\boldsymbol{\mu}}_t \in \Omega$ to the target $\boldsymbol{\nu} \in \Omega$ almost surely and the typical size of the fluctuations of the difference $\hat{\boldsymbol{\mu}}_t - \boldsymbol{\nu}$, where $\Omega = \{\boldsymbol{\nu} \in [0,1]^n \,|\, \sum_{i=1}^n \nu(i) = 1\}$ is the set of all probability distributions on $\mathcal{N}$. If the underlying process on $\mathcal{N}$ were to be a Markov chain, by resorting to similar CLT, we would obtain similar $O(1/\sqrt{t})$ convergence speed where the typical size of the fluctuations is now characterized by that of a n-dimensional multivariate Gaussian random vector with some $n \times n$ covariance matrix. However, the underlying process Z_t in our framework is history-dependent and thus non-Markovian, so we would not expect anything faster than $O(1/\sqrt{t})$ for its error as seen from Markovian or *i.i.d.* cases. Nonetheless, the figures drawn on a log-log scale in our setting demonstrate that the error decays still polynomially fast $O(t^{-\beta})$. These observations are further supported by the mathematical results in [10], which apply to the static NMMC method. Specifically, Corollary 1.3 in [10] implies that if one considers a class of polynomial weight functions $w_k = k^\alpha$, $\alpha \geq 0$, then the difference $\hat{\boldsymbol{\mu}}_t - \boldsymbol{\nu}$ in the l^1 norm is bounded above by $Ct^{-\theta}$ almost surely for some (random) constant $C > 0$, where $\theta \leq \frac{1}{2}$.[6] In addition, Theorem 1.4 in [10] shows that the CLT from $\hat{\boldsymbol{\mu}}_t$ to $\boldsymbol{\nu}$ holds with respect to a *rescaled* time $\eta_t(t)^{-1} \sim t$, as it would hold for Markovian or *i.i.d.* cases.

We now turn our attention to the simulation results under Slashdot and Wiki-Talk graphs. As the graphs are much bigger than the Gnutella graph, we properly increase the number of agents

[6] In other words, for every possible sample path ω of the underlying stochastic process $\{Z_t\}$ by the crawler, there exists some random constant $C(\omega)$ such that the difference $\hat{\boldsymbol{\mu}}_t - \boldsymbol{\nu}$ in the l^1 norm is bounded above by $C(\omega)t^{-\theta}$ almost surely. Note also that θ depends on the extremal value of the spectral functional, which requires the knowledge of the entire spectrum of the underlying graph structure. We refer to [10] for more details.



(a) $\boldsymbol{\pi} = \mathbf{u}$; $p = 0.01$

(b) $\pi(i) \propto d_i^-$; $p = 0.01$

(c) EVC $\mathbf{x}$; $p = 0.01$

Fig. 5. The TVD results (on a log-log scale) obtained under Wiki-Talk graph.

from 100 to 2000. We observe the similar trends as in the Gnutella graph and only present the TVD results for the cases with $w_k = k^\alpha$ ($\alpha = 0, 1, 3, 5, 10$) when using the dynamic NMMC method with c_t-updating probability $p = 0.01$ in Figures 4 and 5, which consistently outperforms the other choices of $p = 1, 0.1$ for all three cases of the target QSD. We still observe a polynomial decrease in the convergence speed of $\hat{\boldsymbol{\mu}}_t$, which is asymptotically linear on a log-log scale. We see that the case with $\alpha = 3$ or $\alpha = 5$ exhibits the fastest decaying rate in its TVD curve for achieving $\boldsymbol{\pi} = \mathbf{u}$ under both Slashdot and Wiki-Talk graphs and for estimating the EVC $\mathbf{x}$ under Slashdot graph. On the other hand, the case with $\alpha = 1$ becomes the best with the fastest decaying rate for the other cases. This again shows that to strike a balance between diffusion and redistribution, the extent of diffusion, which can be controlled by the choice of α, should be properly chosen based on the underlying graph structure and the target QSD. We also note that the convergence becomes much faster when the target QSD is set to be the EVC $\mathbf{x}$ of $\mathbf{A}$ or the in-degree distribution, i.e., $\pi(j) \propto d_j^-$, as compared to the case of the uniform distribution $\boldsymbol{\pi} = \mathbf{u}$.

We finally report the TVD results of the dynamic NMMC method with c_t-updating probability $p = 0.01$ under Amazon graph for the cases with $w_k = k^\alpha$ ($\alpha = 0, 1, 3, 5, 10$) when achieving $\pi(i) \propto d_i^-$. Since the size of the LSCC of the Amazon graph is over ×100 bigger than that of the Gnutella graph, we use 10^4 agents. We again observe the polynomial decay in the convergence speed of $\hat{\boldsymbol{\mu}}_t$. We also see that the case with $w_k = 1$ or $w_k = k$ leads to the fastest decaying rate. This suggests that too strong a bias in weights toward more recently-visited nodes can have an adversary effect, leading to an imbalance between diffusion and redistribution.

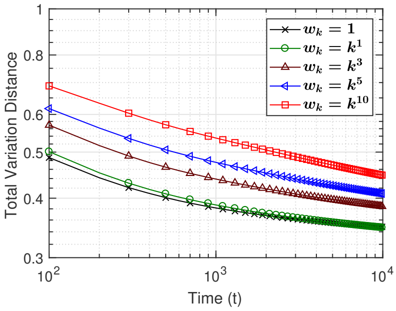


Fig. 6. The TVD results (on a log-log scale) for achieving $\pi(i) \propto d_i^-$ with $p = 0.01$ under Amazon graph.

6 DISCUSSION AND PRACTICAL CONSIDERATIONS

In this section, we thoroughly discuss the comparison between our NMMC method and the relevant literature, ranging from the MCMC methods to the PageRank algorithm. We also address practical issues with our NMMC method and demonstrate that our NMMC method is readily applicable to most direct graphs in practice.

6.1 MCMC vs. NMMC

We first discuss the key differences between the popular MCMC method and our NMMC method for sampling from a given probability distribution $\boldsymbol{\pi}$ on a graph $\mathcal{G}$. In essence, the MCMC method is about how to build a Markov chain $\mathbf{P}$ on a given state space (e.g., graph) such that it attains any desired stationary distribution $\boldsymbol{\pi}$, i.e., $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ on the same space [47, 60]. The Metropolis-Hastings (MH) algorithm, undeniably the most famous one in the MCMC literature, can be described as follows. For a given target distribution $\boldsymbol{\pi}$ to achieve on a graph, consider an underlying Markov chain X_t on this graph with its transition matrix $\mathbf{Q} = [Q_{ij}]$. At the current state $X_t = i$, the chain proposes to move to $X_{t+1} = j$ with probability Q_{ij}, which is then accepted with probability

$$\gamma_{ij} = \min\left\{1, \frac{\pi(j)Q_{ji}}{\pi(i)Q_{ij}}\right\},$$

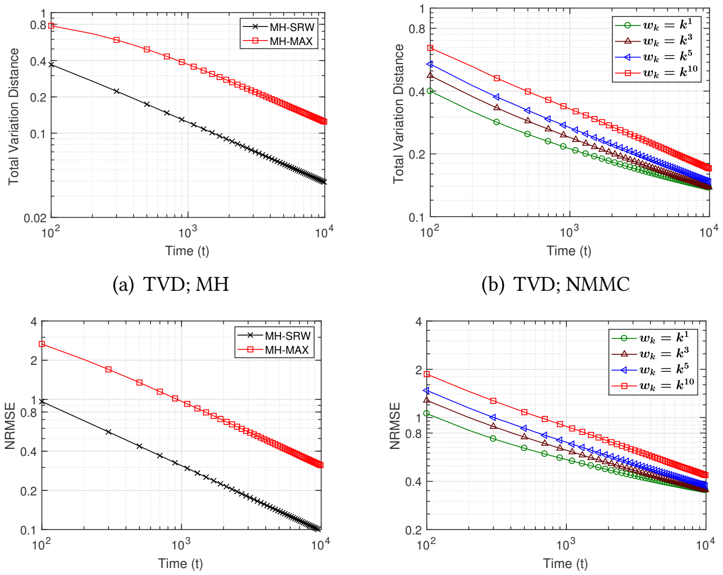
and rejected with probability $1 - \gamma_{ij}$ in which case the chain stays at the same state, i.e., $X_{t+1} = i$. Clearly, it then immediately follows that

$$\pi(i)\gamma_{ij}Q_{ij} = \pi(j)\gamma_{ji}Q_{ji}, \tag{25}$$

for all $i, j \in \mathcal{N}$, i.e., the resulting stochastic matrix $\mathbf{P} = [P_{ij}]$ with $P_{ij} = \gamma_{ij}Q_{ij}$ for $i \neq j$ and $P_{ii} = 1 - \sum_{k \neq i} P_{ik}$ becomes reversible with respect to $\boldsymbol{\pi}$, thus satisfying the original equation $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$. This method has proven to be extremely versatile and powerful, using only local information to find γ_{ij} with any transition matrix Q_{ij} on the given state space. From an algebraic point of view, this amounts to constructing a reversible *stochastic* matrix $\mathbf{P}$ such that its leading left-eigenvector matches a given $\boldsymbol{\pi}$. As mentioned in Section 1, again, the most important yet implicit assumption in (25) is that $Q_{ji} > 0$ whenever $Q_{ij} > 0$, i.e., nodes i and j have reciprocal relationship. If not, i.e., there exists a directed edge from i to j but not vice versa, we would have $Q_{ji} = 0$ even if $Q_{ij} > 0$, for which the MH algorithm no longer applies. What we have established in this paper is to show how we can remove the requirements of (i) being stochastic and (ii) reciprocity in the underlying process.

One may ask what if our NMMC method is applied to an 'undirected' graph and/or how it would be compared to the MH algorithm under the undirected graph for graph sampling. We again emphasize that our NMMC method is the first of its kind, applicable to directed graphs or in general when the proposed move is non-reciprocal. Our NMMC method is not meant to be compared with the MH algorithm, as the MH cannot even be applied to directed graphs and our algorithm is not an improvement over the MH algorithm.

Nonetheless, for the sake of completeness, we provide simulation results of the MH algorithm and our NMMC method when applied to an undirected graph. To this end, we use the undirected version of the Gnutella graph (its strongly connected component). For the MH algorithm, we consider the following two popular random-walk versions, since it still depends on the choice of $\mathbf{Q}$. One is to use $Q_{ij} = 1/d_{\max}$ if $(i,j) \in \mathcal{E}$ and $Q_{ij} = 0$ if $(i,j) \notin \mathcal{E}$ and $i \neq j$, with $Q_{ii} = 1 - d_i/d_{\max}$, where d_i is the degree of node i and $d_{\max}$ is the maximum degree of the undirected graph [6, 14]. We call this version MH-MAX. The other is to use the simple random walk for $\mathbf{Q}$, where $Q_{ij} = 1/d_i$ if $(i,j) \in \mathcal{E}$ and $Q_{ij} = 0$, otherwise [32, 44, 65]. We call this version MH-SRW. Note that the transition matrix of the simple random walk is still used as the proposed chain $\mathbf{Q}$ under our NMMC method, but the resulting acceptance probability in (10) is now given by $\gamma_{ij} \propto \frac{\pi(j)}{\pi(i)}\frac{d_i}{d_j}$, since the underlying graph is *undirected.*



(a) TVD; MH

(b) TVD; NMMC

(c) NRMSE; MH

(d) NRMSE; NMMC

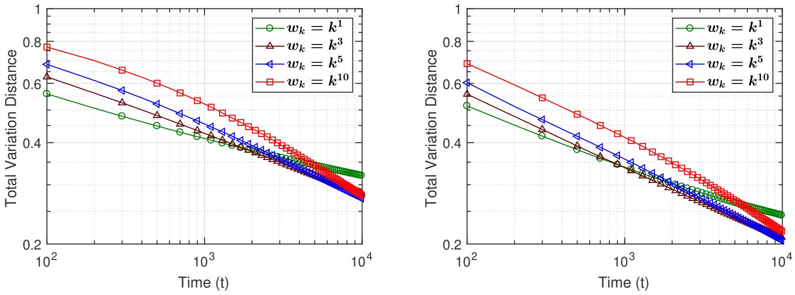
Fig. 7. The TVD and NRMSE results (on a log-log scale) of (a) MH-SRW and MH-MAX and (b) NMMC method with $w_k = k^\alpha$ ($\alpha = 3, 5, 10$) and $p = 0.01$ when applied to an *'undirectified'* version of Gnutella graph.

We here consider $\boldsymbol{\pi} = \mathbf{u}$ and use 100 random walks. Letting $\{X_t\}$ be the trajectory by the MH-MAX (or MH-SRW), we consider $\hat{x}(i) := \frac{1}{t}\sum_{k=1}^{t} \mathbf{1}_{\{X_k = i\}}$ to estimate $\pi(i) = 1/n$. For performance metrics, we use the TVD and also the normalized root mean square error (NRMSE), which is defined as $\sqrt{\mathbb{E}\{(\hat{x}(i) - 1/n)^2\}}/(1/n)$ for each i. We report the average of the NRMSE over all $i \in \mathcal{N}$. Figure 7 shows the TVD and NRMSE results (on a log-log scale) of MH-MAX, MH-SRW, and our NMMC method with $w_k = k^\alpha$ ($\alpha = 1, 3, 5, 10$) and $p = 0.01$. We see that all the TVD and NRMSE values decay to zero polynomially fast as we discussed in Section 5.

6.2 PageRank vs. NMMC

Our NMMC method may be reminiscent of the famous PageRank algorithm and its associated 'random surfer' model due to the seemingly common nature of following successive outgoing links at random and occasionally performing a random jump to a node in the graph. They are, however, fundamentally different, as explained below.

The random surfer model was introduced in the seminal work by Brin and Page [16, 54] in order to *interpret* the PageRank vector as the stationary distribution, say $\boldsymbol{\pi}^*$, of an 'imaginary' random walk (or random surfer) on a Web graph. Namely, a random surfer keeps clicking successive outgoing links uniformly at random and occasionally gets bored of following links then jumps or teleports to a random page (by entering a new destination in the browser's URL line). In the long run, the proportion of the time the random surfer spends on a given page becomes the PageRank score of the page. We again emphasize that the random surfer model was just a random-walk "interpretation" of the PageRank algorithm, *not* an actual algorithmic operation implementable by actual crawlers on the graph. The rationale behind such a random-walk interpretation is to enable one to resort to the results of random walks on a graph, or the theory of Markov chains, in order to understand the properties of the PageRank algorithm such as convergence properties as in the seminal work [16, 54] and follow-up studies [39, 43].



(a) $\boldsymbol{\pi} = \mathbf{u}; p = 0.1$ (b) $\boldsymbol{\pi} = \mathbf{u}; p = 0.01$

Fig. 8. The TVD results (on a log-log scale) obtained under Gnutella graph for the NMMC method with an online estimate of the in-degree of each node $i \in \mathcal{N}$.

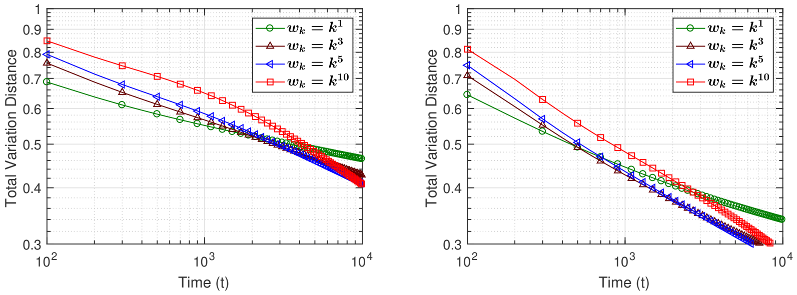
However, we are not concerned with speeding up the PageRank algorithm nor devising an efficient numerical method for computing $\boldsymbol{\pi}^*$. We are rather focused on a somewhat reverse problem, which is to obtain samples from a desired probability distribution $\boldsymbol{\pi}$ over a directed graph without requiring the global knowledge of the graph structure. It is also worth noting that unlike the operation of a random surfer/walk in the random surfer model that is to teleport to *any* random node (even including not-yet-visited ones), our NMMC method is only limited to making a random jump (as a process of redistribution) to one of *already-visited* nodes, which can be readily implemented in practice without acquiring the global knowledge. We do not assume a full access to the state space, which would be necessary to realize a teleportation in the random surfer model if one wishes to jump to any arbitrary page (associated with its own URL) in a Web graph, and would be equivalent to assuming that one can draw *i.i.d.* samples directly from the space.

6.3 Online In-degree Estimation

We validate the feasibility of our NMMC method with an online estimate of the in-degree of each node, which is possibly latent information. As explained in 3.2, the in-degree information is often readily available as part of a user profile in online social networks such as Google++, Twitter, and Instagram. However, in other types of directed graphs like Web graphs, the in-degree information may not be available but needs to be discovered. To make our NMMC method applicable to the latter, we use a simple way of estimating the in-degree of node i. Each crawling agent, in the stage of proposing a movement from i to j, can discover the presence of an incoming edge from i into j. Whenever a new incoming edge is discovered, the agent increments the value of the in-degree of j, say $\hat{d}_j^-$, with $\hat{d}_j^- := 1$ initially. All the agents share the estimate of the in-degree of each node. We report the resulting performance of our NMMC method in Figure 8 and observe that our NMMC method performs well even with the simple online in-degree estimation. This manifests the robustness and versatility of our NMMC method in practice.

6.4 Going Beyond Strongly Connected Directed Graphs

We have assumed that the underlying graph $\mathcal{G}$ is strongly connected so far. While this assumption has been made mainly for ease of exposition, it may not hold in practice. We thus discuss how we can relax this assumption so that our NMMC method becomes ready for practical use. We first note that the key assumption for the NMMC method is essentially the irreducibility of the 'proposed' Markov chain $\mathbf{Q}$ on transient states $\mathcal{N}$, as explained in Section 3. This may not be equivalent to assuming that the underlying directed graph be strongly connected for sampling from an arbitrary probability distribution $\boldsymbol{\pi}$. It is worth noting that for the estimation of the EVC $\mathbf{x}$ of the directed



(a) $\boldsymbol{\pi} = \mathbf{u}; p = 0.1$

(b) $\boldsymbol{\pi} = \mathbf{u}; p = 0.01$

Fig. 9. The TVD results (on a log-log scale) of our NMMC method on the set of nodes $\mathcal{N}_S$, which are reachable from any node in the LSCC of Gnutella graph.

graph, the underlying graph needs to be strongly connected by definition, or to properly define the eigenvector centrality as a valid measure [50]. In this case, the assumption of strongly connected graphs is not a limitation at all. Thus, we below focus only on achieving $\boldsymbol{\pi}$.

If the directed graph $\mathcal{G}$ is not strongly connected, the support of $\boldsymbol{\pi}$ would need to be on the set of all the nodes, say $\mathcal{N}_S$, that are "reachable" by traversing directional edges when starting from an initial node, or a set of initial nodes, say S, that are reachable among themselves. Note that the "unreachable" nodes $\mathcal{N} \setminus \mathcal{N}_S$ are simply invisible and can safely be ignored. The technical changes are then how to deal with (i) 'dangling' nodes, which are the nodes with no forward links, and (ii) a set of nodes that are reachable but without outgoing edges to the other nodes in $\mathcal{N}_S$. Note that these two problems were resolved in the PageRank algorithm by adopting the teleportation to any node in the graph, yet to numerically compute the (unknown) PageRank scores. While such global teleportation is infeasible for sampling in practice, its *personalized* PageRank version is practically implementable and can be adopted into our NMMC method.[7]

Fix the set of initial nodes S. The state space $\mathcal{N}$ then reduces to $\mathcal{N}_S$. For the proposed chain $\mathbf{Q}$ on $\mathcal{N}_S$, we consider the following transition matrix: For $i, j \in \mathcal{N}_S$,

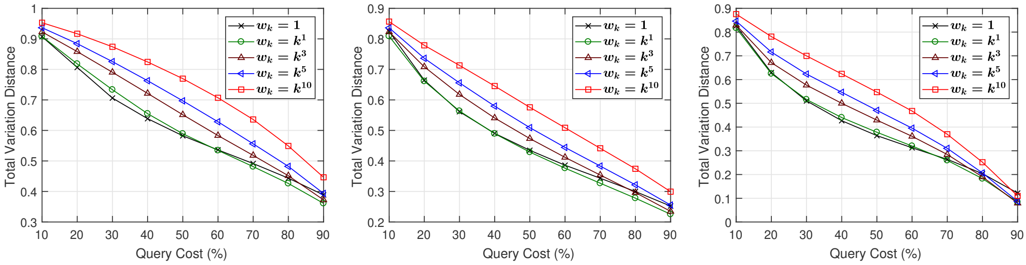
$$Q_{ij} = (1-p)\frac{1}{|S|} + p\frac{1}{|S|}\mathbf{1}_{\{d_i^+=0\}} + p\frac{A_{ij}}{d_i^+}\mathbf{1}_{\{d_i^+>0\}}, \tag{26}$$

for some constant $p \in (0,1)$. At each node i, with probability p, the agent follows one of its outgoing edges (if exists) uniformly at random, and jumps to one of the nodes in S uniformly with probability $1-p$. If node i is a dangling node with no outgoing edges ($d_i^+ = 0$), then the agent always jumps to a random node in S. It is not difficult to see the irreducibility of $\mathbf{Q}$ on $\mathcal{N}_S$ for our NMMC method. Then, the acceptance probability γ_{ij} in (9) can properly be obtained. In Figure 9, we present the TVD results for estimating the uniform $\mathbf{u}$ on $\mathcal{N}_S$ using the dynamic NMMC method with $\mathbf{Q}$ in (26), where $\mathcal{N}_S$ is the set of nodes that are reachable from any node in the LSCC of Gnutella graph and $|\mathcal{N}_S| = 8,566$. We here use 300 independent crawling agents starting from 300 fixed, distinct nodes S and $p = 0.95$. Figure 9 demonstrates the feasibility of our NMMC method even when the underlying graph is not strongly connected.

6.5 Query Cost vs. Performance for Graph Sampling

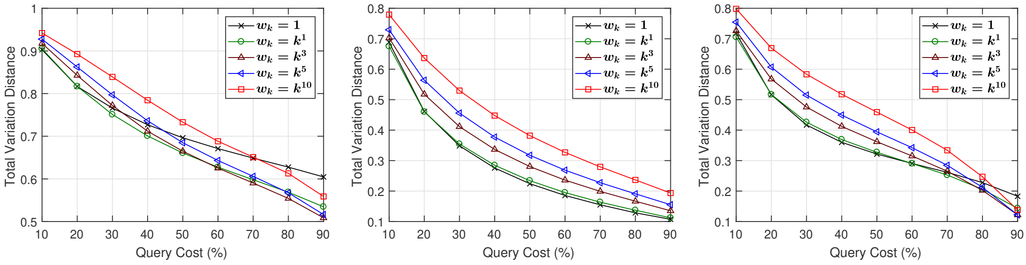
So far, all the simulation results have been shown with respect to the number of time steps, since they have the primary purpose of numerically validating the convergence of our NMMC method

[7] The personalized PageRank scores are still unknown and need to be computed by the PageRank algorithm (or the power method), as long as the PageRank scores themselves are concerned.



(a) $\boldsymbol{\pi} = \mathbf{u}; p = 0.01$ (b) $\pi(i) \propto d_i^-; p = 0.01$ (c) EVC $\mathbf{x}; p = 0.01$

Fig. 10. The TVD results obtained under Gnutella graph as the query cost varies.



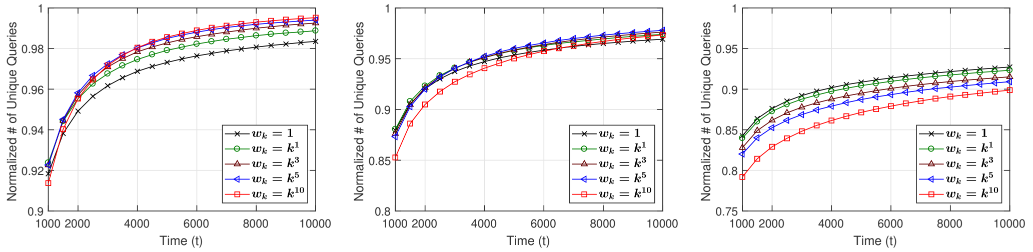
(a) $\boldsymbol{\pi} = \mathbf{u}; p = 0.01$ (b) $\pi(i) \propto d_i^-; p = 0.01$ (c) EVC $\mathbf{x}; p = 0.01$

Fig. 11. The TVD results obtained under Slashdot graph with varying degrees of the query cost.

and evaluating its convergence speed. Recall that the ergodic theorem in (12) and the CLT in (24) hold for ergodic Markov chains with respect to the number of time steps. Nonetheless, when it comes to graph sampling via crawling, 'query cost' becomes practically important. In general, crawling real-world networks is done through public yet restrictive API interfaces provided by the networks or their Web interfaces (by following URLs and hyperlink structures). Such external access is often rate-limited due to security concerns, privacy protection, infrastructure burden and commercial purposes [73, 77]. For example, Twitter allows only 15 API requests to retrieve IDs of a user's followers or IDs of the friends that the specific user is following every 15 minutes [69].

From a practical point of view, it would be informative to see the performance of our NMMC method with respect to the query cost in terms of the API calls (or HTTP requests) required. We below present the performance of NMMC method with respect to the query cost under Gnutella and Slashdot graphs. To this end, we adopt the definition of the query cost commonly used in the literature [32, 59, 78, 79], which is the number of *unique* queries required. In other words, API calls are spent only when visiting a node (or querying its page/profile) *for the first time*, and revisiting the same node since then (duplicate queries) can be done immediately by retrieving the nodal information from local cache without consuming the query rate limit. This indicates that the redistribution mechanism of our NMMC method is achieved at no additional cost as long as query cost is concerned. Thus, the 'due' redistributions are only advantageous to shape up the historical empirical distribution $\hat{\boldsymbol{\mu}}_t$ toward the target QSD. The local cache can also be easily shared among multiple crawling agents.

In Figures 10 and 11, we present the TVD results of the dynamic NMMC method with k^α ($\alpha = 0, 1, 3, 5, 10$) and $p = 0.01$ under Gnutella and Slashdot graphs, respectively, as the query cost varies. Here, the query cost is defined as the number of unique queries normalized by the graph size (LSCC size) and given in percentage. The same numbers of agents are used as before for Figures 2



(a) $\boldsymbol{\pi} = \mathbf{u}$; $p = 0.01$ (b) $\pi(i) \propto d_i^-$; $p = 0.01$ (c) EVC $\mathbf{x}$; $p = 0.01$

Fig. 12. The normalized number of unique queries used by the dynamic NMMC method on Gnutella graph.

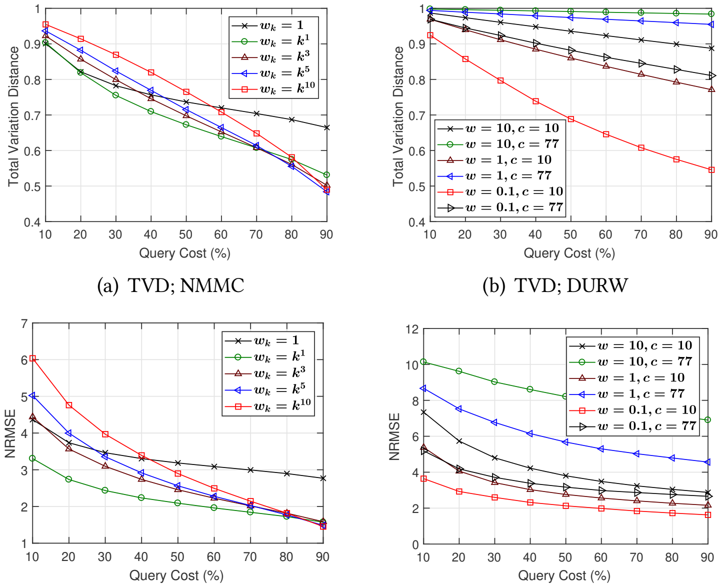
and 4, respectively. We see that the choice of $w_k = k$ exhibits the overall fastest decaying rate in its TVD curve for all three cases of the target QSD under both Gnutella and Slashdot graphs. This also indicates that higher redistribution weights toward more recently-visited nodes is no longer beneficial in improving the convergence speed when the query cost is concerned.

In a similar vein, we showed the TVD results of our NMMC method until 10^4 time steps in Figure 2, which is greater than the LSCC size of the Gnutella graph. Considering 100 agents used, one might expect that the entire graph is already visited by that time. However, it has not been the case, since there are a lot of redistributions that are counted toward the number of time steps, but not toward the number of unique queries. The multiple agents can also visit the already-visited nodes many times, which is again not counted toward the number of unique queries. This is confirmed in Figure 12, which shows the number of unique queries, which is normalized by the LSCC size, for all three cases of the target QSD on the Gnutella graph using the dynamic NMMC method with k^α $(\alpha = 0, 1, 3, 5, 10)$ and $p = 0.01$.

6.6 Comparison with the Latest Approach for Graph Sampling

Lastly, we provide a comparison with the latest approach in [59] for sampling directed graphs. Among only a few studies available in the literature on sampling from directed graphs, which shall be explained later in Section 7, the latest practical solution is the one in [59], which is called Directed Unbiased Random Walk (DURW) algorithm. While our NMMC method is capable of achieving any distribution $\boldsymbol{\pi}$ on a directed graph and estimating its EVC $\mathbf{x}$, which is much more than just achieving 'uniform' samples from the directed graph, the comparison of our NMMC method to the DURW algorithm would be interesting in the context of uniform graph sampling and is provided below. We here focus on estimating $\boldsymbol{\pi} = \mathbf{u}$.

The DURW algorithm has two main parts. The first part is to transform the underlying directed graph to an *undirected* graph. At each newly visited node, it keeps outgoing edges only to 'not-yet-visited' nodes and converts them as undirected edges. This edge-pruning idea was originally proposed in [6]. Upon the transformed, undirected graph, the second part is to use (i) a weighted random walk with random jumps to *everywhere* in the graph and (ii) the importance sampling. Specifically, the weighted random walk at node i chooses one of its neighbors uniformly at random, with probability $\frac{d_i}{d_i+w}$, and jumps (or teleports) to any random one in the graph, with probability $\frac{w}{d_i+w}$, where w is the weight parameter for choosing a random jump and d_i is the degree of i. While the former operation consumes a unit cost if a new node is sampled, the latter consumes the cost $c > 1$. For a directed social network, assuming that its user-ID space is accessible, the cost c of a random jump is the number of API calls required on average to find a valid ID for sampling. It clearly depends on how sparse the user-ID space may be, which is unknown and uncontrollable. Note that the random jumps may not be feasible at all in practice, e.g., for sampling Web graphs.



(a) TVD; NMMC

(b) TVD; DURW

(c) NRMSE; NMMC

(d) NRMSE; DURW

Fig. 13. The TVD and NRMSE results of our dynamic NMMC method and the DURW algorithm under Gnutella graph.

In addition, letting $\boldsymbol{\pi}'$ be the stationary distribution of the weighted random walk with random jumps, one can see that $\pi'(i)$ is proportional to $d_i + w$. The importance sampling is then applied to correct the bias from the non-uniform $\boldsymbol{\pi}'$. Thus, letting $\{X_t\}$ be the trajectory by the DURW algorithm, it is to use the ratio estimator $\hat{x}(i) := \frac{\sum_{k=1}^{t} \mathbf{1}_{\{X_k = i\}}/\pi'(X_k)}{\sum_{k=1}^{t} 1/\pi'(X_k)}$ to estimate $\boldsymbol{\pi} = \mathbf{u}$. Note that, instead of $\boldsymbol{\pi}'$, its unnormalized version can be used in practice. We refer to [59] for more details.

Figure 13 shows the TVD and NRMSE results of our NMMC method with $w_k = k^{\alpha}$ ($\alpha = 0, 1, 3, 5, 10$) and $p = 0.01$, and the DURW algorithm with various choices of w and c under Gnutella graph, which are from [59]. For example, the choice of $w = 0.1$ behaves more like the pure simple random walk, while the larger values of w put more weights on random jumps. In addition, $c = 10$ means 10 API calls spent on average to sample a valid ID (or to teleport to a random node). Similarly for $c = 77$. In addition, we here consider a single random crawler. It is because in the DURW algorithm, each crawler has to build its own state space (or the transformed, undirected graph) as it explores the graph, which is clearly time-varying and depends on its sample path, hence the state space can never be shared with other crawlers. The query cost is again defined as the number of unique queries normalized by the graph size (LSCC size) and given in percentage. We report the average of the NRMSE over all $i \in \mathcal{N}$.

We see that our NMMC method generally performs better than the DURW algorithm. A possible explanation would be that the way to transform the underlying directed graph to an undirected graph makes the transformed one 'tree-like'. Thus, traversing the transformed graph could get slower. While random jumps can speed up the graph traversal, the high cost of each random jump eventually outweighs the benefit of moving to everywhere in the graph. We also observe that carefully choosing the value of w with given (yet unknown) cost c, the DURW algorithm would be comparable to our NMMC method. In other words, the DURW algorithm requires a good choice of w a priori against unknown c, before running the random crawler to explore a directed graph by ‘undirectifying’ directed edges on the fly. In contrast, our NMMC method performs well for a wide range of values of α for $w_k = k^{\alpha}$.

We finally emphasize that while we take (directed) graph sampling as a primary application of our NMMC method in this paper, the NMMC method can potentially find much broader scope of applications beyond the graph sampling as we have witnessed the widespread applications of the MH algorithm in many disciplines. The potential broader impact stems from the versatile applicability of the NMMC method for generating *any* arbitrary probability distribution over the state spaces violating the reciprocity condition, where the MH algorithm and other standard MCMC methods are no longer applicable.

7 RELATED WORK

There have been a few studies on achieving uniform samples from *directed* graphs, ranging from traditional Web graphs [6, 40, 61] to recent online social networks [30, 59, 75]. The key underlying methods have been mainly (explicitly or implicitly) the MH algorithm and importance sampling for Markov chains as in (22). The ‘non-uniform’ random walks, along with the importance sampling, are typically the simple random walks and (personalized) PageRank walks [40, 61]. Unfortunately, however, their stationary distributions are simply *unknown*, and the only viable approach in [40, 61] has been to empirically “estimate" the stationary distribution (e.g., based on the visit counts obtained by running a separate random walk), which is still known to suffer from the bias toward the nodes with high in-degree [7]. Another simple way to get around the difficulty is to just convert every directed edge to an undirected one (excluding the duplicate ones) [30, 61, 75], which technically has noting to do with sampling from directed graphs.

A more practical method was proposed in [6] to construct an undirected graph on-the-fly from a target directed graph for sampling, only considering the inbound links for each node discovered before the node is visited for the first time, and then to apply, in essence, the MH algorithm with $Q_{ij} = 1/\max_k d_k^+$ over the transformed *undirected* graph to obtain uniform samples. In short, the edge conversion is done with edge pruning. This idea has been adopted later in [59] for a weighted random walk with ‘random jumps’ (to anywhere in the undirected graph), equipped with the importance sampling due to its non-uniform stationary distribution (on the undirected graph). In contrast to all these, *without* knowing the stationary distributions or requiring any transformation of the directed graph to another undirected version, our NMMC method is able to sample from any target distribution over the directed graph with theoretical guarantees. We recently came to know the work by Avrachenkov et al. [4] in which they briefly mention that their proposed reinforcement learning-based sampling technique can potentially be applied to strongly connected directed graphs, but without any theoretical and practical demonstrations. We here emphasize that our NMMC method is even capable of sampling from a given distribution $\boldsymbol{\pi}$ over the set of reachable nodes $\mathcal{N}_S$ of the directed graph, even if the graph is not strongly connected, not to mention the capability of estimating the *EVC* $\mathbf{x}$ of the (strongly connected) directed graph.

8 CONCLUSION

In this paper, we have shown how to go beyond the limitation set by the current MCMC methods when state spaces violate the reciprocity condition and the global knowledge is unavailable. Unlike the most studies in the literature that aim to improve a given MCMC algorithm or to propose a better one within the same framework, our approach is fundamentally different and unique, in that it works directly over any nonreversible proposed chain such as simple random walks on directed graphs without sacrificing their distributed nature in the implementation. Our NMMC framework builds upon our careful mapping from the proposed chain to the target distribution on a set of transient states, and entails the machinery of the quasi-stationary distribution of a suitably constructed transient chain and the induced random walks with reinforcement to relocate to positions from their past history, which are amenable to distributed implementation using only locally available information. We have also devised a number of ways to improve the overall speed of convergence and to enhance the practicability, by utilizing online estimates of in-degree and the required global constant and by relaxing the assumption of the strong connectedness of the underlying directed graph, to make our NMMC method readily applicable to most directed graphs in reality.

9 ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and especially our shepherd, Konstantin Avrachenkov, for many constructive comments and suggestions that greatly improved the quality of this paper. Do Young Eun is supported in part by National Science Foundation under Grant Nos. CNS-1423151 and CNS-1824518.

REFERENCES

[1] David Aldous, Barry Flannery, and José Luis Palacios. 1988. Two Applications of Urn Processes The Fringe Analysis of Search Trees and The Simulation of Quasi-Stationary Distributions of Markov Chains. *Probability in the Engineering and Informational Sciences* 2, 3 (1988), 293–307.

[2] Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. 2007. Non-Backtracking Random Walks Mix Faster. *Communications in Contemporary Mathematics* 9, 4 (2007), 585–603.

[3] Amine Asselah, Pablo A. Ferrari, and Pablo Groisman. 2011. Quasistationary distributions and Fleming-Viot processes in finite spaces. *Journal of Applied Probability* 48, 2 (Jun 2011), 322–332.

[4] Konstantin Avrachenkov, Vivek S. Borkar, Arun Kadavankandy, and Jithin K. Sreedharan. 2018. Revisiting random walk based sampling in networks: evasion of burn-in period and frequent regenerations. *Computational Social Networks* 5, 1 (Mar 2018), 19.

[5] Konstantin Avrachenkov, Bruno Ribeiro, and Jithin K. Sreedharan. 2016. Inference in OSNs via Lightweight Partial Crawls. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'16)*. 165–177.

[6] Ziv Bar-Yossef, Alexander Berg, Steve Chien, Jittat Fakcharoenphol, and Dror Weitz. 2000. Approximating Aggregate Queries About Web Pages via Random Walks. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB'00)*. 535–544.

[7] Ziv Bar-Yossef and Maxim Gurevich. 2008. Random Sampling from a Search Engine's Index. *J. ACM* 55, 5, Article 24 (Oct 2008), 24:1–24:74 pages.

[8] A. A. Barker. 1965. Monte Carlo Calculations of the Radial Distribution Functions for a Proton-Electron Plasma. *Australian Journal of Physics* 18, 2 (1965), 119–134.

[9] Michel Benaïm. 1997. Vertex-reinforced random walks and a conjecture of Pemantle. *Annals of Probability* 25, 1 (01 1997), 361–392.

[10] Michel Benaïm and Bertrand Cloez. 2015. A stochastic approximation approach to quasi-stationary distributions on finite spaces. *Electronic Communications in Probability* 20 (2015), 1–14.

[11] Michel Benaïm, Bertrand Cloez, and Fabien Panloup. 2018. Stochastic approximation of quasi-stationary distributions on compact spaces and applications. *Annals of Applied Probability* 28, 4 (08 2018), 2370–2416.

[12] Joris Bierkens. 2016. Non-reversible Metropolis-Hastings. *Statistics and Computing* 26, 6 (2016), 1213–1228.

[13] J. Blanchet, P. Glynn, and S. Zheng. 2016. Analysis of a stochastic approximation algorithm for computing quasi-stationary distributions. *Advances in Applied Probability* 48, 3 (09 2016), 792–811.

[14] S. Boyd, P. Diaconis, and L. Xiao. 2004. Fastest mixing markov chain on a graph. *SIAM Rev.* 46, 4 (2004), 667–689.

[15] P. Brémaud. 1999. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues.*

[16] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems* 30, 1-7 (April 1998), 107–117.

[17] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. 2009. Localization of the Maximal Entropy Random Walk. *Physical Review Letters* 102 (Apr 2009), 160602. Issue 16.

[18] Geoffrey S. Canright and Kenth Engø-Monsen. 2006. Spreading on Networks: A Topographic View. *Complexus* 3 (Aug 2006), 131–146. Issue 1-3.

[19] Iacopo Carreras, Daniele Miorandi, Geoffrey S. Canright, and Kenth Engø-Monsen. 2007. *Eigenvector Centrality in Highly Partitioned Mobile Networks: Principles and Applications.* Springer Berlin Heidelberg.

[20] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. 2008. Epidemic Thresholds in Real Networks. *ACM Transactions on Information and System Security* 10, 4 (Jan 2008), 1:1–1:26.

[21] Chen Chen, Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2016. Eigen-Optimization on Large Graphs by Edge Manipulation. *ACM Transactions on Knowledge Discovery from Data* 10, 4 (Jun. 2016), 49:1–49:30.

[22] Fang Chen, László Lovász, and Igor Pak. 1999. Lifting Markov Chains to Speed Up Mixing. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing (STOC'99)*. 275–281.

[23] Ting-Li Chen and Chii-Ruey Hwang. 2013. Accelerating reversible Markov chains. *Statistics & Probability Letters* 83, 9 (2013), 1956–1962.

[24] Persi Diaconis, Susan Holmes, and Radford M. Neal. 2000. Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability* 10, 3 (Aug 2000), 726–752.

[25] Persi Diaconis and Laurent Miclo. 2013. On the spectral analysis of second-order Markov chains. *Annales de la Faculté des sciences de Toulouse : Mathématiques* 22 (2013), 573–621. Issue 3.

[26] Persi Diaconis and Laurent Saloff-Coste. 1998. What Do We Know about the Metropolis Algorithm? *J. Comput. System Sci.* 57, 1 (1998), 20–36.

[27] Moez Draief, Ayalvadi Ganesh, and Laurent Massoulié. 2008. Thresholds for Virus Spread on Networks. *Annals of Applied Probability* 18, 2 (04 2008), 359–378.

[28] David Easley and Jon Kleinberg. 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press.

[29] Heitor C.M. Fernandes and Martin Weigel. 2011. Non-reversible Monte Carlo simulations of spin models. *Computer Physics Communications* 182, 9 (2011), 1856–1859.

[30] Maksym Gabielkov, Ashwin Rao, and Arnaud Legout. 2014. Sampling Online Social Networks: An Experimental Study of Twitter. In *Proceedings of ACM SIGCOMM.*

[31] Ayalvadi Ganesh, Laurent Massoulié, and Don Towsley. 2005. The Effect of Network Topology on the Spread of Epidemics. In *Proceedings of IEEE INFOCOM.*

[32] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. 2011. Practical Recommendations on Crawling Online Social Networks. *IEEE Journal on Selected Areas in Communications* 29, 9 (October 2011), 1872–1892.

[33] Ilie Grigorescu and Min Kang. 2004. Hydrodynamic limit for a Fleming-Viot type system. *Stochastic Processes and their Applications* 110, 1 (2004), 111–143.

[34] Ilie Grigorescu and Min Kang. 2012. Immortal particle for a catalytic branching process. *Probability Theory and Related Fields* 153, 1 (2012), 333–361.

[35] Ilie Grigorescu and Min Kang. 2013. Markov processes with redistribution. *Markov Process and Related Fields* 19, 3 (2013), 497–520.

[36] Pablo Groisman and Matthieu Jonckheere. 2013. Simulation of Quasi-Stationary Distributions on Countable Spaces. *Markov Processes and Related Fields* 19, 3 (2013), 521–542.

[37] Stephen J. Hardiman and Liran Katzir. 2013. Estimating Clustering Coefficients and Size of Social Networks via Random Walk. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 539–550.

[38] W. K. Hastings. 1970. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* 57, 1 (1970), 97–109.

[39] Taher Haveliwala and Sepandar Kamvar. 2003. *The Second Eigenvalue of the Google Matrix.* Technical Report 2003-20. Stanford University.

[40] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. 2000. On Near-uniform URL Sampling. *Computer Networks* 33, 1 (2000), 295–308.

[41] Akihisa Ichiki and Masayuki Ohzeki. 2013. Violation of detailed balance accelerates relaxation. *Physical Review E* 88 (Aug 2013), 020101. Issue 2.

[42] Marcus Kaiser, Robert L. Jack, and Johannes Zimmer. 2017. Acceleration of Convergence to Equilibrium in Markov Chains by Breaking Detailed Balance. *Journal of Statistical Physics* 168, 2 (Jul 2017), 259–287.

[43] Amy N. Langville and Carl D. Meyer. 2006. *Google's PageRank and Beyond: The Science of Search Engine Rankings.* Princeton University Press.

[44] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2012. Beyond Random Walk and Metropolis-Hastings Samplers: Why You Should Not Backtrack for Unbiased Graph Sampling. In *Proceedings of the ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'12)*. 319–330.

[45] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2017. On the Rao-Blackwellization and Its Application for Graph Sampling via Neighborhood Exploration. In *Proceedings of IEEE INFOCOM.*

[46] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data/.

[47] Jun S. Liu. 2004. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag.

[48] Linyuan Lu, Duanbing Chen, Xiao-Long Ren, Qian-Ming Zhang, Yi-Cheng Zhang, and Tao Zhou. 2016. Vital nodes identification in complex networks. *Physics Reports* 650 (2016), 1–63.

[49] Sylvie Méléard and Denis Villemonais. 2012. Quasi-stationary distributions and population processes. *Probability Surveys* 9 (2012), 340–410.

[50] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra.* SIAM.

[51] Radford M. Neal. 2004. *Improving Asymptotic Variance of MCMC Estimators: Non-reversible Chains are Better.* Technical Report No. 0406. Department of Statistics, University of Toronto.

[52] M. E. J. Newman. 2003. The Structure and Function of Complex Networks. *SIAM Rev.* 45, 2 (2003), 167–256.

[53] M. E. J. Newman. 2010. *Networks: An Introduction.* Oxford University Press.

[54] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford University.

[55] Robin Pemantle. 2007. A survey of random processes with reinforcement. *Probability Surveys* 4 (2007), 1–79.

[56] P. H. Peskun. 1973. Optimum Monte-Carlo Sampling Using Markov Chains. *Biometrika* 60, 3 (1973), 607–612.

[57] Luc Rey-Bellet and Konstantinos Spiliopoulos. 2016. Improving the Convergence of Reversible Samplers. *Journal of Statistical Physics* 164, 3 (Aug 2016), 472–494.

[58] Bruno Ribeiro and Don Towsley. 2010. Estimating and Sampling Graphs with Multidimensional Random Walks. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC'10).* 390–403.

[59] Bruno Ribeiro, Pinghui Wang, Fabricio Murai, and Don Towsley. 2012. Sampling Directed Graphs with Random Walks. In *Proceedings of IEEE INFOCOM.*

[60] Matthew Richey. 2010. The Evolution of Markov Chain Monte Carlo Methods. *The American Mathematical Monthly* 117, 5 (2010), 383–413.

[61] Paat Rusmevichientong, David M. Pennock, Steve Lawrence, and C. Lee Giles. 2001. Methods for Sampling Pages Uniformly from the World Wide Web. In *Proceedings of AAAI Fall Symposium on Using Uncertainty Within Computation.* 121–128.

[62] Yuji Sakai and Koji Hukushima. 2016. Eigenvalue analysis of an irreversible random walk with skew detailed balance conditions. *Physical Review E* 93 (Apr 2016), 043318. Issue 4.

[63] Raoul D. Schram and Gerard T. Barkema. 2015. Monte Carlo methods beyond detailed balance. *Physica A: Statistical Mechanics and its Applications* 418 (2015), 88–93.

[64] Roberta Sinatra, Jesús Gómez-Gardeñes, Renaud Lambiotte, Vincenzo Nicosia, and Vito Latora. 2011. Maximal-entropy random walks in complex networks with limited information. *Physical Review E* 83 (Mar 2011), 030103. Issue 3.

[65] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. 2009. On Unbiased Sampling for Unstructured Peer-to-peer Networks. *IEEE/ACM Transactions on Networking* 17, 2 (Apr 2009), 377–390.

[66] Yi Sun, Juergen Schmidhuber, and Faustino J. Gomez. 2010. Improving the Asymptotic Performance of Markov Chain Monte-Carlo by Inserting Vortices. In *Advances in Neural Information Processing Systems 23.* 2235–2243.

[67] G. Timár, A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes. 2017. Mapping the Structure of Directed Networks: Beyond the Bow-Tie Diagram. *Physical Review Letters* 118 (Feb 2017), 078301. Issue 7.

[68] Konstantin S. Turitsyn, Michael Chertkov, and Marija Vucelja. 2011. Irreversible Monte Carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena* 240, 4 (2011), 410–414.

[69] Twitter. [n. d.]. Rate Limiting. https://dev.twitter.com/rest/public/rate-limiting.

[70] Erik A. van Doorn and Philip K. Pollett. 2009. Quasi-stationary distributions for reducible absorbing Markov chains in discrete time. *Markov Process and Related Fields* 15, 2 (2009), 191–204.

[71] Erik A. van Doorn and Philip K. Pollett. 2013. Quasi-stationary distributions for discrete-state models. *European Journal of Operational Research* 230, 1 (2013), 1–14.

[72] Piet Van Mieghem, Jasmina Omic, and Robert Kooij. 2009. Virus Spread in Networks. *IEEE/ACM Transactions on Networking* 17, 1 (Feb 2009), 1–14.

[73] Norases Vesdapunt and Hector Garcia-Molina. 2016. Updating an Existing Social Graph Snapshot via a Limited API. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16).* 1693–1702.

[74] Marija Vucelja. 2016. Lifting–A nonreversible Markov chain Monte Carlo algorithm. *American Journal of Physics* 84, 12 (2016), 958–968.

[75] Tianyi Wang, Yang Chen, Zengbin Zhang, Peng Sun, Beixing Deng, and Xing Li. 2010. Unbiased Sampling in Directed Social Graph. In *Proceedings of ACM SIGCOMM.*

[76] Xin Xu, Chul-Ho Lee, and Do Young Eun. 2014. A General Framework of Hybrid Graph Sampling for Complex Network Analysis. In *Proceedings of IEEE INFOCOM.*

[77] Xin Xu, Chul-Ho Lee, and Do Young Eun. 2017. Challenging the Limits: Sampling Online Social Networks with Cost Constraints. In *Proceedings of IEEE INFOCOM.*

[78] Zhuojie Zhou, Nan Zhang, and Gautam Das. 2015. Leveraging History for Faster Sampling of Online Social Networks. *Proceedings of the VLDB Endowment* 8 (Jun. 2015), 1034–1045. Issue 10.

[79] Zhuojie Zhou, Nan Zhang, Zhiguo Gong, and Gautam Das. 2016. Faster Random Walks by Rewiring Online Social Networks On-the-Fly. *ACM Transactions on Database Systems* 40 (Jan. 2016), 26:1–26:36. Issue 4.