SIAM J. Sci. Comput.
Vol. 32, No. 5, pp. 2937–2963

SCALABLE UNIFORM GRAPH SAMPLING BY LOCAL COMPUTATION[*]

SATU ELISA SCHAEFFER[†]

Abstract. We address the problem of obtaining by local computation a sample of a graph such that the vertices are sampled uniformly. The presented solution uses a Markov chain to combine a rapidly mixing random walk that does not reach a uniform distribution with a slow-mixing walk that does. The resulting chain mixes at a notably faster rate than the standard uniform random walk and can be simulated exactly using only local information.

Key words. Markov chain, sampling, uniform distribution, rapid mixing, graph, random walk

AMS subject classifications. 60J20, 60J22, 62D05

DOI. 10.1137/080716086

1. Introduction.

Random sampling is a powerful tool in the construction of efficient algorithms for demanding computational problems [12, 13, 26, 34, 35]. Sampling methods are useful, for example, in lossy data compression [4]. They also help us to analyze and understand properties of large combinatorial objects.

The order and complexity of systems studied by computational science, such as communication networks and genomic data, increase, and often the size of the data sets available surpass the limits of scalability of existing tools for analyzing them. *Sampling* provides a way to analyze properties of data sets that are too large to process as is: with a sampling method, one obtains a subset of the data that should be representative of the whole data set, but of much smaller order.

In the case of graphs, obtaining a subset of vertices can be helpful in estimating the structural properties of a network. For example, calculating the average path length is computationally demanding for a large graph, but computing a set of shortest paths among a smaller set of vertices can be done to obtain an estimate of the global average. The caveat is making sure that the sample is not distorted, i.e., that it actually reflects the properties of the source data from which it was obtained. A good example of using a uniform random sample to estimate an important graph property, namely, the betweenness centrality (cf. [15, 28]) is the algorithm of Eppstein and Wong [22]; another related work is that of Newman [42], which is based on random walks.

This is the fundamental question of sampling: How do we select the subset to examine so that the estimates obtained reflect the global properties of the graph as faithfully as possible, while keeping the size of the sample needed relatively small?

[*] Received by the editors February 19, 2008; accepted for publication (in revised form) July 16, 2010; published electronically September 29, 2010. The author was supported by SEP-PROMEP of Mexico under grants 103,5/07/2523 and 103,5/08/4804. The early stages of this work were done in collaboration with Pekka Orponen at Helsinki University of Technology (TKK) under grant 206235 (ANNE 2003–2006) of the Academy of Finland, and parts of that work have been reported in the doctoral dissertation of the author [47]. An early version of this work was submitted to the arXiv.org e-Print archive as a part of a two-topic work [43].

http://www.siam.org/journals/sisc/32-5/71608.html

[†] Center for Innovation, Research, and Development in Engineering and Technology (CIIDIT) & Department of Electrical and Mechanical Engineering (FIME), Universidad Autónoma de Nuevo León (UANL), Av. Universidad s/n, Cd. Universitaria, San Nicolás de los Garza, N.L. 66450, Mexico (elisa.schaeffer@uanl.edu.mx).

It is often practical to assume that if a sample of a graph reflects some commonly used structural properties, such as the clustering coefficient, average path length, and degree distribution (cf. [20]), then it can also be used to estimate other properties, although counterexamples of samples that preserve certain properties while losing others are easy to construct. Repeating the sampling and using more than one sampling method can be employed to improve the reliability of the observations made. The need for repetitions, however, makes it even more important that the sampling procedure does not consume much computation or memory.

Typically, what one aims for is a sample that include all vertices of the graph with the same probability, regardless of their degrees. Such sampling is called *uniform sampling* of the graph. If the number of vertices, together with an ordering on the vertex set, are known, it is trivial to obtain a uniform random sample by simply taking the list of vertices and picking numbers from $[1, 2, \ldots, n]$ uniformly at random to select the sample.

However, some graphs of interest are massive, rapidly changing, or in other aspects infeasible for obtaining global snapshots or calculating the exact vertex count. In order to uniformly sample such a graph, *local computation* offers a scalable approach: the order of the graph should not affect the sampling procedure. The graph structure should be explored progressively without needing to read large portions of the graph into main memory, preferably without needing access to all of the graph at any time.

In this paper we concentrate on sampling undirected, unweighted graphs. Extensions to weighted and directed cases are left to future work; such methods are of great interest and are necessary for numerous applications, such as sampling the Web graph [8, 36], which is a directed graph representing the pages and hyperlinks on the World Wide Web. On the other hand, bioinformatics applications often involve weighted graphs [32, 51, 56].

The rest of the paper is organized as follows. First, in section 2, we establish the notation and terminology necessary throughout the paper. In section 3, we review some of the related work. Then we proceed to define Markov chains for graph sampling in section 4. After presenting the constructions and their formal proofs, we explain the implementation and the experimental evaluation in section 5, and then conclude the work in section 6.

2. Background.

In order to discuss sampling in mathematical terms, basic knowledge of stochastic processes is necessary, especially regarding discrete-time processes operating in a discrete state space. For readers unfamiliar with the fundamentals of the topic, we recommend a comprehensive text book, such as that of Grimmett and Stirzaker [29].

Let $G = (V, E)$ be an undirected, unweighted connected graph. Let $v, w \in V$ be *vertices* of the graph and E the set of undirected *edges*. We often enumerate the vertices notationally as in $V = \{v_1, v_2, \ldots, v_n\}$ and associate the vertex v_i with its index i. If $(v, w) \in E$, we say that the two vertices are *neighbors*. The *degree* of a vertex is the number of neighbors it has. The essential graph-theoretical notation we use is summarized in Table 1.

Let the vertex set V be the state space of a discrete, time-homogeneous Markov chain with time steps $t = 0, 1, 2, \ldots$. As the graph is connected and undirected, defining the transition probabilities $p_{v,w}$ as being strictly positive for each edge $(v, w) \in E$ (and zero elsewhere), the resulting chain is irreducible and all the states are positive recurrent. Hence, the chain necessarily has one or more stationary distributions π_s to which the chain converges into a stationary distribution. The stationary distri-

Table 1
Notation used in this paper.

$G = (V, E)$	undirected, unweighted connected graph
V	set of *vertices*
E	set of *edges*
n	number of vertices, $\lvert V\rvert$
m	number of edges, $\lvert E\rvert$
$\Gamma(v)$	set of *neighbors* of $v \in V$
$\deg(v)$	*degree* of vertex v, $\lvert\Gamma(v)\rvert$

bution of a Markov chain that is both irreducible and aperiodic is unique (that is, independent of the initial distribution); such chains are called ergodic.

For an irreducible Markov chain that has *only* positive recurrent states, there exists a *reversed* chain. Denoting the irreducible positive recurrent Markov chain by $\mathbf{M} = \{M_\ell : \ell \in 1, 2, \ldots, t\}$, the reversed chain is $\mathbf{M}^{\text{rev}}_\ell = M_{t-\ell}$. The transition probabilities of the reversed chain are

$$\Pr[\mathbf{M}^{\text{rev}}_{t+1} = w \mid \mathbf{M}^{\text{rev}}_t = v] = \frac{\pi_s(w)}{\pi_s(v)} \cdot p_{w,v}, \tag{2.1}$$

in terms of the transition probabilities and the stationary distribution of the original chain $\mathbf{M}$.

The *detailed balance* conditions

$$\forall v,\, w \in V : \pi_s(v) \cdot p_{v,w} = \pi_s(w) \cdot p_{w,v} \tag{2.2}$$

hold for a stationary distribution of a reversible Markov chain. Also, irreversible chains can have stationary distributions, although the detailed balance conditions do not hold. Reversibility of a chain $\mathbf{M}$ implies that the two chains $\mathbf{M}$ and $\mathbf{M}^{\text{rev}}$ are statistically indistinguishable at equilibrium.

The number of steps it takes a given Markov chain started with an arbitrary initial distribution to converge to the stationary distribution is of special interest; it is called the *mixing time* Φ_ϵ of the chain and is defined, in terms of a parameter $\epsilon > 0$, as the earliest time step after which $\mathcal{D}(\pi, \pi_s) \le \epsilon$ holds for all future time steps, where

$$\mathcal{D}(\pi_1, \pi_2) = \frac{1}{2} \sum_{v \in V} \lvert \pi_1(v) - \pi_2(v) \rvert \tag{2.3}$$

is the *total variation distance* (TVD) between two distributions π_1 and π_2. In rough terms, if the chain reaches stationary distribution in a number of steps that is polynomial in the input size (i.e., order of the state space) and also polynomial in $\frac{1}{\epsilon}$, the process is said to be *rapidly mixing.*

The *eigenvalue spectrum* of the transition matrix can be used to evaluate the mixing time of the chain. The primary eigenvalue λ_1 of a stochastic matrix is one. The Perron–Frobenius theorem [29] states that for the nonprincipal eigenvalues, $\lvert\lambda_i\rvert \le \lambda_1 = 1$. If the eigenvalue one has a multiplicity greater than one or there are complex roots that lie on the unit circle, the chain is reducible and has more stationary distributions.

As any vector, including the initial distribution, can be represented as an eigenvalue decomposition in the vector space determined by the eigenvectors, and all λ_i other than those corresponding to stationary distributions have absolute value smaller than one, the corresponding components get smaller and smaller as the chain is run. This implies that the smaller the eigenvalues λ_i are, the faster the chain converges to the stationary distribution [33]. In particular, knowing the second eigenvalue λ_2 (in decreasing order of absolute value) allows us to characterize the mixing time as

$$\mathcal{O}\left(\frac{\log(n)}{1-|\lambda_2|}\right), \tag{2.4}$$

which shows that the smaller the second eigenvalue is, the faster the chain will mix [49]. For more information on mixing times, we recommend the books by Behrends [6] and Sinclair [50].

3. Related work. As a first approach to sampling a graph, one would consider picking any vertex of the graph and performing a *random walk* from that initial vertex: in each step, a neighbor of the current vertex is chosen to be the next vertex. After a sufficient number of steps, one would expect to have wandered sufficiently far from the initial vertex to claim to be at a "randomly chosen" vertex. This random walk is known as the *regular* random walk (also called *simple* or *blind* random walk).

Note that for general graphs, this chain does *not* converge to a uniform stationary distribution, although there are special cases where it does, such as degree-regular graphs (where the distribution is evidently uniform, as all the degrees are equal) and expander graphs (where the distribution is a constant factor away from being uniform [27]). Also note that, whereas all chains corresponding to regular random walks on connected undirected graphs are irreducible Markov chains, they may well be periodic, for which the general case may not converge to a unique stationary distribution.

As an example of a large undirected graph, we mention the Internet. In 1997, Paxson and Floyd [46] identified the difficulty in characterizing Internet topology and attributed it to the constant change and growth of the network. Their research on Internet simulation is motivated by the possibility of approaching "complicated scenarios that would be either difficult or impossible to analyze" [46]. In 2003, Floyd and Kohler [25] discussed the need for better models of the Internet. Now, several years later, the size of the network is already significantly bigger and the problems related to its topology are urgent. Innovative and adaptive routing protocols are needed for the ever-growing amount of traffic and the imbalance of the routing load—much of the traffic flows along a few prevalent routes instead of spreading throughout the available infrastructure [45], the configuration of the routers is complex [24], and the current solutions are not considered sufficiently scalable [37]. Due to these research challenges, characterizing and estimating the structural properties is of great interest.

Krishnamurthy et al. [38] discuss sampling of the Internet, with an emphasis on how small a sample can be and still be useful and informative. It would be helpful to be able to predict the future evolution of the network in order to design better hardware, traffic protocols, and routing algorithms [23, 54]. For a broader discussion of modeling the Internet as well as the World Wide Web, we recommend the book by Baldi, Frasconi, and Smythe [3].

Returning to the theme of random walks, as the random regular walk follows any edge outward from the current vertex with equal probability, the stationary distribution favors vertices of high degree, and hence any sampling done by a regular random walk will be skewed towards the hubs[1] of a nonuniform network. The measurements

[1] A hub is a vertex with a particularly high degree in comparison to the average degree of the graph.

made on the degree distribution of the Internet suffered from a similar bias, although instead of sampling single vertices, shortest paths between pairs were sampled [14]. Achlioptas et al. [2] show that such path-based sampling can make even a Poissonian degree distribution seem scale-free, as well as a uniform distribution (i.e., a regular graph). An analysis on what causes such a bias is provided by Dall'Asta et al. [16].

The case of vertex sampling is, however, resolvable. There are several options available for enhancing the regular random walk to obtain a uniform sample over vertex degrees. A relatively simple method is to apply *rejection sampling*, accepting a sample with a probability proportional to the inverse of the degree of the sampled vertex [52]. Possible problems include the difficulty of estimating the proportion of acceptable samples in the set of samples obtained, and hence uncertainty of the running time of the method for a given number of samples needed. Also there are some mathematical constraints on when such a construction is feasible [11].

Another possibility is to add a sufficient number of *reflexive edges* to each vertex that does not have the maximum degree D and create a modified multigraph G' that is regular with the maximum degree of the original graph [18]. This means that each vertex $v \in V$ of the original graph G is included in G', but with $D - \deg(v)$ (directed) self-loops included in the edge set in addition to the original edges in E. As each vertex in G' has the same degree, they all have equal probability in the distribution of equation (4.2) for the modified graph G'. Intuitively, such a walk on G' will stall on an originally low-degree vertex for a long time, whereas it passes through high-degree vertices much more quickly.

While simulating such a walk, considering all the self-loops separately can be avoided by making following observation: a self-loop will be chosen with probability

$$q = \frac{D - \deg(v)}{D}, \tag{3.1}$$

which enables the algorithm to "flip a coin" to determine whether to stay in the same state or to take an outbound transition. This is essentially a Bernoulli trial with success probability $q' = 1 - q$, where success is interpreted as choosing to follow a transition that *leaves* the current state. This observation enables us to avoid actually having to construct G' with such a large number of edges, as the expected number of Bernoulli trials before a success is geometrically distributed with parameter q'. Hence one simply needs to draw a geometrically distributed random number to obtain the number of steps that the chain should stall at that state, and then continue with the transition probabilities of the regular random walk on the original graph G. We call this construction, which uses the geometrically distributed random variable to sample the graph, the *coin-flip* random walk.

In this paper, we propose a combination of two random walks into a single Markov chain: we combine the regular random walk with another that converges to the uniform distribution, and then we sample from the chain at specific times to ensure a uniform sample. The construction will be explained in detail in the next section.

A walk-combination approach similar to ours has been taken by Wei, Erenrich, and Selman [55] in sampling satisfiable evaluations for a SAT (satisfiability) instance. They use a random walk mixed with the Walk-SAT [48] algorithm[2] to obtain a near-uniform sample of satisfiable truth assignments, whereas using Walk-SAT alone would

[2] Walk-SAT is a popular satisfiability solver introduced in mid-1990s, i.e., an algorithm to find satisfying truth assignments for the variables of a logical formula.

result in a nonuniform sample. We expect our combined chain to provide a starting point for similar constructions.

Also Datta and Kargupta [17] use Markov chains to obtain uniform samples of peer-to-peer (P2P) networks. The P2P networks are represented as undirected graphs, where the vertices represent the data rather than the peers. The structure of such systems coincides with the so-called power-law graphs that naturally permit a rapid mixing [40]. Our construction differs from the work of Datta and Kargupta in the sense that our approach can be applied to improve the mixing towards also other stationary distributions and not just the uniform distribution—by changing the sampling side probabilities and appropriately adjusting the side-switching probabilities to fulfill the detailed balance conditions, a different distribution is achieved.

4. Markov chain constructions. In this section we define the three random walks discussed in this paper and derive the stationary distribution of each chain.

4.1. Regular random walk. Formally, a regular random walk on a graph is a Markov chain in which the transition probabilities out of a vertex are uniform:

$$p_{v,w} = \begin{cases} \dfrac{1}{\deg(v)}, & w \in \Gamma(v), \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The stationary distribution of such a chain is

$$\pi_r = (\pi_r(v_1), \ldots, \pi_r(v_n)) = \left(\frac{\deg(v_1)}{2m}, \ldots, \frac{\deg(v_n)}{2m}\right), \tag{4.2}$$

as $2m$ is the total number of edge endpoints in G. The distribution of (4.2) can be shown to be a stationary distribution by studying the detailed balance conditions in equilibrium, as their validity implies stationarity for a distribution:

$$\begin{aligned} \pi_r(v) \cdot p_{v,w} &= \pi_r(w) \cdot p_{w,v}, \\ \frac{\deg(v)}{2m} \cdot \frac{1}{\deg(v)} &= \frac{\deg(w)}{2m} \cdot \frac{1}{\deg(w)}. \end{aligned} \tag{4.3}$$

As the equality holds, global equilibrium follows from the local equilibrium of the detailed balance conditions, and the distribution is stationary.

4.2. Balanced random walk. One can also define a Markov chain that has the uniform distribution as the stationary distribution. This is achieved by choosing transition probabilities [7]:

$$p_{v,w} = \begin{cases} \min\left\{\frac{1}{\deg(v)}, \frac{1}{\deg(w)}\right\} & \text{if } w \in \Gamma(v), \\ 1 - \sum_{w \in \Gamma(v)} \min\left\{\frac{1}{\deg(v)}, \frac{1}{\deg(w)}\right\} & \text{if } w = v, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

We call this the *balanced* random walk.[3] Note that the transition matrix is symmetric and hence doubly stochastic. Another noteworthy property is that the presence of

[3] A simpler version with these same properties is to use $p_{v,w} = (\deg(v) \cdot \deg(w))^{-1}$ for $w \in \Gamma(v)$, zero for nonneighbors, and the rest on a self-loop, but the self-loop probability often turns out to be impractically large and slows down the mixing of the walk.

just one nonzero diagonal term suffices to guarantee that the walk is aperiodic, given that the graph is connected.

Due to symmetry with respect to v and w, detailed balance trivially holds also for this definition with the uniform distribution as the stationary distribution:

$$\frac{1}{n}\min\left\{\frac{1}{\deg(v)},\frac{1}{\deg(w)}\right\}=\frac{1}{n}\min\left\{\frac{1}{\deg(w)},\frac{1}{\deg(v)}\right\}. \tag{4.5}$$

The self-loop probability $p_{v,v}$ can be rewritten in a simpler form: if $\deg(v)\geq\deg(w)$, the subtracted term is always $\deg(v)^{-1}$. Otherwise, the subtraction is *smaller* by $\deg(v)^{-1}-\deg(w)^{-1}$. Hence, if we sum over $w\in\Gamma(v)$, the self-loop probability is the sum of these leftovers. Using $\deg(v)=|\Gamma(v)|$, we obtain

$$\begin{aligned} p_{v,v} &= 1-\sum_{w\in\Gamma(v)}\min\left\{\frac{1}{\deg(v)},\frac{1}{\deg(w)}\right\}\\ &= \sum_{w\in\Gamma(v)}\left(\frac{1}{\deg(v)}-\min\left\{\frac{1}{\deg(v)},\frac{1}{\deg(w)}\right\}\right)\\ &= \sum_{w\in\Gamma(v)}\max\left\{\frac{1}{\deg(v)}-\frac{1}{\deg(v)},\frac{1}{\deg(v)}-\frac{1}{\deg(w)}\right\}\\ &= \sum_{w\in\Gamma(v)}\max\left\{0,\frac{1}{\deg(v)}-\frac{1}{\deg(w)}\right\}. \end{aligned} \tag{4.6}$$

Intuitively, a walk that visits the hubs of a nonuniform network can quickly reach any part of the network. Continuing that line of thought, chains such as this balanced walk that *avoid* visiting hubs take a longer time to cover the whole graph. Hence, we expect the balanced chain to mix poorly. Later in section 5 we discuss the eigenvalue spectra of the regular random walk transition matrix (equation (4.1)) and the balanced random walk transition matrix (equation (4.4)).

4.3. Combined random walk. In order to construct a rapidly mixing Markov chain for uniform sampling of $G=(V,E)$, we create a *"mirror vertex"* v' for each vertex $v\in V$, connect the original vertex and the mirror vertex to each other by transitions, and use different transition probabilities between the original vertices than those we use with the mirrors.[4] We call such a chain a *combined random walk*. A small example of the mirror construction is illustrated in Figure 1.

The original vertices $v\in V$ of the input graph G are called the *sampling side* of the modified graph and the mirror vertices $v'\in V'$ form the *mixing side*. The goal of the construction is to ensure that each transition probability can be computed *locally* in the neighborhood of the current state based on the *adjacency list* of the vertex corresponding to the current state and the *degrees* of the neighbors.[5] Keeping the computation as local as possible is desirable to improve the scalability: for large and dynamic graphs, no global information is available, and even estimates on figures such as the graph order, size, or maximum degree can be time-consuming or impractical to obtain.

[4] The goal is to locally compute the transition probabilities based on the degrees of the neighbors, as detailed later in this article.

[5] In the usual sense, local computation only involves knowledge on the identities of the neighbor nodes and not on their degrees. In our case, the locality involves this lookahead: it is necessary to traverse the neighbor list of each neighbor in order to determine its degree.

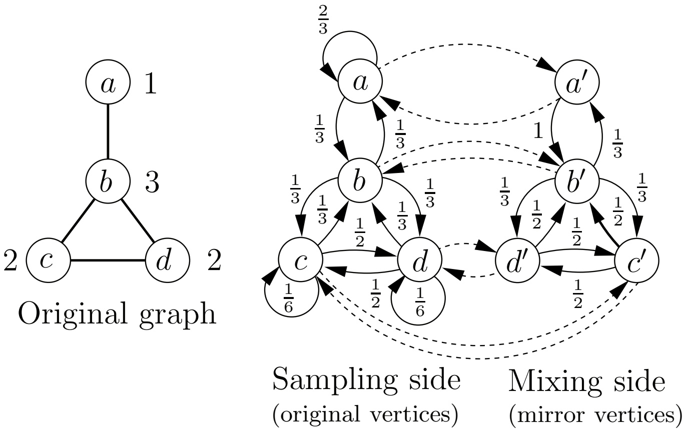

FIG. 1. *Left: An example graph where $V = \{a, b, c, d\}$ with the degree of each vertex shown beside it. Right: A schematic construction of the Markov chain of the combined random walk* without *having yet taken aside the probabilities ε and ε' that are illustrated as the dotted-line transitions—the values shown are the transition probabilities of the balanced (on the sampling side) and the regular (on the mixing side) chains. The self-loops have been omitted from the figure for clarity.*

The transition probabilities on the sampling side are set relative to those of either of the above balanced random walks, and on the mixing side, a regular random walk is mimicked with minor modifications. For transitions from a vertex v to its mirror vertex v', we set

$$p_{v,v'} = \varepsilon, \tag{4.7}$$

where ε is a parameter of the construction. Hence we need to set aside probability mass on the sampling side in order to ensure that

$$\varepsilon + p_{v,v} + \sum_{w \in \Gamma(v)} p_{v,w} = 1 \tag{4.8}$$

for each vertex v on the sampling side. The balanced chain has self-loops, but unfortunately $p_{v,v}$ are arbitrarily close to zero, for example in the presence of a large, star-topology induced subgraph. Hence we need to design such variations of the chains to be sure that $p_{v,v} \geq \varepsilon$ on the walk on the original graph, such that the probability ε may be subtracted from the self-loop when constructing the combined chain, without altering the other transition probabilities between vertices on the sampling side. Note that for $\varepsilon > 0$, the combined chain is ergodic, which guarantees the existence of a unique stationary distribution.

We achieve such a design by introducing a guaranteed-weight self-loop for each vertex on the sampling side. For example, if we want to ensure that $\varepsilon \geq \frac{1}{2}$, we divide each transition probability out of each vertex by two and add the $\frac{1}{2}$ thus gained (as the initial outgoing flow was necessarily one and was halved) to the self-loop probability. For any $\gamma > 1$, we may thus ensure that $\varepsilon \geq \frac{1}{\gamma}$. For a fixed γ, the transition probabilities $p_{v,w}$ of a given Markov chain are modified as follows to allow $p'_{v,v} \geq \gamma^{-1}$, and hence loosening the restrictions on ε for all $v \in V$:

$$p'_{v,w} = \begin{cases} \dfrac{\gamma - 1}{\gamma} \cdot p_{v,w} & \text{if } w \in \Gamma(v), \\ \dfrac{\gamma - 1}{\gamma} \cdot p_{v,w} + \dfrac{1}{\gamma} & \text{if } v = w. \end{cases} \tag{4.9}$$

Applying such a modification to the balanced chain, we arrive at the following transition probabilities:

$$p_{v,w} = \begin{cases} \frac{\gamma-1}{\gamma}\min\left\{\frac{1}{\deg(v)}\ \frac{1}{\deg(w)}\right\} & \text{if } w \in \Gamma(v), \\ \frac{1}{\gamma} + \sum_{w\in\Gamma(v)} \frac{\gamma-1}{\gamma}\max\left\{0,\ \frac{1}{\deg(v)} - \frac{1}{\deg(w)}\right\} & \text{if } v = w, \\ 0 & \text{otherwise,} \end{cases} \tag{4.10}$$

where we continue to denote by $\deg(v)$ the degree of v *in the original graph* G, i.e., ignoring in the degree the added edge that connects the two sides.

Similarly, we add a self-loop to each vertex $v' \in V'$ such that

$$p_{v',v'} \geq \varepsilon'_v, \tag{4.11}$$

where ε'_v is the probability of returning to the sampling side from v',

$$p_{v',v} = \varepsilon'_v. \tag{4.12}$$

See Figure 2 for an illustration of the connections between the sampling and the mixing sides. For the example graph of Figure 1, the resulting Markov chain would contain eight states, two per each vertex, a pair of additional transitions between each vertex and its copy on the mixing side (the dotted arrows of Figure 1), two transitions per each edge of the input graph, and a self-loop per each vertex, giving a total of 24 transitions. The probabilities associated with these transitions depend on the value chosen for ε.

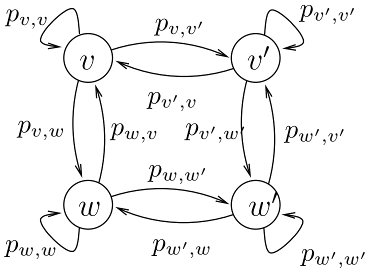

Fig. 2. *A diagram of the mirror construction for two vertices v and w on the sampling side and their mirror vertices v' and v' on the mixing side.*

The transition probability from the sampling side to the mixing side is constant over the vertices on the sampling side, but the return probability from the mixing side varies depending on the mirror vertex, as will be shown later. Hence we need to fix a probability

$$\delta \geq \max_{v'} \varepsilon'_v \tag{4.13}$$

for the self-loop probability on the mixing side so that we can always safely subtract ε'_v from the self-loop of the mixing side chain. The transition probabilities within the mixing side are therefore

$$p_{v',w'} = (1-\delta)\frac{1}{\deg(v)}. \tag{4.14}$$

The stationary distribution π_c of the combined chain is a weighted combination of the distributions of the sampling side, π_b, and that of the mixing side, π_r, such that an α-fraction of the time, the Markov chain is on the sampling side, and a $(1-\alpha)$-fraction of the time is spent on the mixing side:

$$\begin{aligned}\pi_c &= (\pi_c(v_1), \ldots, \pi_c(v_n), \pi_c(v_1'), \ldots, \pi_c(v_n')) \\ &= \big(\alpha\pi_b(v_1), \ldots, \alpha\pi_b(v_n), (1-\alpha)\pi_r(v_1), \ldots, (1-\alpha)\pi_r(v_n)\big).\end{aligned} \tag{4.15}$$

THEOREM 4.1. *For any $\alpha \in (0,1)$ and $\varepsilon \in (0,1)$, there exists an ε_v' for each $v \in V$ such that* (4.16) *is a stationary distribution for the combined chain for the choice of ε and α.*

The relationship between the values of ε and α is derived in the proof. Before entering the details of the proof, we wish to highlight the fact that the values of α and ε will fix the values that need to be set for ε_v', as will be shown in the proof. Also, the reason that ε must be strictly positive is that it guarantees the aperiodicity of the combined chain, and hence the existence of a unique stationary distribution.

Proof. We again examine the detailed balance conditions (equation (2.2)) for the above distribution to show that it is a stationary distribution of the combined chain. There are three cases to consider; for a self-loop, the detailed balance conditions trivially hold by definition regardless of the transition probabilities.

1. Transitions within V on the sampling side: $v \leftrightarrow w$:
   As the transition probabilities $p_{v,w}$ in V for $v \neq w$ have not changed other than the introduction of the same multiplicative constant $\frac{\gamma-1}{\gamma}$ for ensuring the self-loop to be able to cover for ε, we now multiply both sides of (4.4) by the multiplicative constants $\frac{\gamma-1}{\gamma}$ and α, which both cancel out.
2. Transitions within V' on the mixing side: $v' \leftrightarrow w'$:
   The transition probabilities together with the above distribution fulfill the detailed balance conditions, as we need only add the multiplicative coefficients $(1-\alpha)$ and $(1-\delta)$ on each side of (4.3), and they cancel out.
3. Transitions between V and V': $v \leftrightarrow v'$:
   This condition can be met by setting dependencies between the parameters of the construction, the transition probability ε, the return probabilities ε_v', and the weighting coefficient α that determines the proportion of time spent on the sampling side, as described below.

The detailed balance condition of the third type is

$$\begin{aligned}\pi_c(v) \cdot p_{v,v'} &= \pi_c(v') \cdot p_{v',v}, \\ \alpha \cdot \frac{1}{n} \cdot \varepsilon &= (1-\alpha) \cdot \frac{\deg(v)}{2m} \cdot \varepsilon_v'.\end{aligned} \tag{4.16}$$

From their definitions we know that $\alpha \in (0,1)$, $\varepsilon \in (0,1)$, and, for all $v \in V$, also $\varepsilon_v' \in (0,1)$. Solving the above equation for ε_v' gives

$$\varepsilon_v' = \frac{2m\alpha\varepsilon}{n(1-\alpha)\deg(v)} = \frac{2m}{n} \cdot \frac{\alpha}{(1-\alpha)} \cdot \varepsilon \cdot \deg(v)^{-1}. \tag{4.17}$$

Using the above value for ε_v' fulfills the last of the three detailed balance conditions and completes the proof. ∎

The first coefficient $\frac{2m}{n}$ is the average degree k of the original graph G. This is a global property of the graph, but we can eliminate its presence in the parameter equation by further restricting α such that

$$\frac{\alpha}{1-\alpha} \cdot k = 1 \Rightarrow \alpha = \frac{1}{k+1}. \tag{4.18}$$

A pleasant feature of the construction is that α does not need to be known to implement the combined chain: it influences only the portion of time the chain spends on the sampling side. Hence knowing α is useful for determining the number of steps that the chain needs to be run until it converges. If no information on the average degree is available a priori, one can obtain an estimate *during* the sampling walk itself by taking the average degree of the vertices that have been visited on the sampling side of the construction. This estimate should improve as the chain converges—we experimentally examine this in section 5 (Figure 13).

Using (4.18), equation (4.17) becomes simply

$$\varepsilon'_v = \frac{\varepsilon}{\deg(v)}, \tag{4.19}$$

which in turn gives us a safe value for δ, as ε'_v is maximized for the minimum degree in G, which in a connected graph is always at least one, giving $\delta \geq \varepsilon$. Setting $\delta = \varepsilon$ we eliminate the presence of the δ-parameter in the construction of the combined chain.

The above construction of combining two chains—one rapidly mixing but to an undesired distribution and the other slowly mixing but to a distribution of interest—can be generalized to scenarios other than uniform sampling. The prerequisite is the introduction of a self-loop on both sides such that the side-change transitions can be subtracted without affecting the relative stationary probabilities on each side. Also, to achieve the detailed balance conditions for the crossings from one side to another, one needs to define ε'_v of a vertex v to fulfill (4.16),

$$\alpha \cdot \pi_1(v) \cdot \varepsilon = (1-\alpha) \cdot \pi_2(v) \cdot \varepsilon'_v, \tag{4.20}$$

where π_1 is the stationary distribution of the sampling side chain alone, and π_2 is that of the mixing side chain. This property would allow applying the construction to problems in combinatorial generation, enumeration, and counting, where a problem instance needs to be obtained according to some distribution of interest, but all simple-definition chains converging to the distribution in question mix impractically slowly.

Analytically, the improvement in the mixing time can be argued as follows. Let λ_i^R be eigenvalues of the regular walk and λ_i^B those of the balanced walk. The two $n \times n$ transition matrices—denote them by $\mathbf{P}_R$ and $\mathbf{P}_B$—have the same nonzero pattern, with the exception of the diagonal, which is zero for the regular walk and can take positive values on the balanced walk. The effect of the diagonal on the spectrum of a transition matrix is simple: the heavier the diagonal, the larger the value of the second eigenvalue. This agrees with the intuition on self-loops slowing down the mixing.

Denoting a unit vector by $\mathbf{1} = (1, 1, \ldots, 1)$, the transition matrix $\mathbf{P}_C$ of the combined chain is a $2n \times 2n$ matrix, where the upper left quadrant is $(1-\varepsilon)\mathbf{P}_B$, the upper right quadrant is $\operatorname{diag}(\varepsilon\mathbf{1})$, the lower left is $\operatorname{diag}(\mathbf{e})$ where $\mathbf{e} = (\varepsilon'_1, \varepsilon'_2, \ldots, \varepsilon'_n)$, and the lower right quadrant is $\operatorname{diag}(\mathbf{1}-\mathbf{e})\mathbf{P}_R$:

$$(4.21)\quad P_C = \left(\begin{array}{ccc|ccc} (1-\varepsilon)p^{(B)}_{1,1} & \vdots & (1-\varepsilon)p^{(B)}_{1,n} & \varepsilon & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \varepsilon & \vdots \\ (1-\varepsilon)p^{(B)}_{n,1} & \ldots & (1-\varepsilon)p^{(B)}_{n,n} & 0 & \ldots & \varepsilon \\ \hline \varepsilon'_1 & \ldots & 0 & (1-\varepsilon'_1)p^{(R)}_{1,1} & \ldots & (1-\varepsilon'_1)p^{(R)}_{1,n} \\ \vdots & \varepsilon'_k & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \varepsilon'_n & (1-\varepsilon'_n)p^{(R)}_{n,1} & \ldots & (1-\varepsilon'_n)p^{(R)}_{n,n} \end{array}\right).$$

Now, if we were to set $\varepsilon = 0$ and $\varepsilon'_i = 0$ for all i, the spectrum of $\mathbf{P}_C$ would simply be a list of all the λ^B_i and λ^R_i combined (corresponding to a *reducible* Markov chain with two eigenvalues equal to one). As we have already observed, adding probability mass on the diagonal by assigning $\varepsilon > 0$, causing all the ε'_i's to be nonzero as well, makes the chain irreducible and aperiodic, dropping a principal eigenvalue just below one. However, λ^C_2 of $\mathbf{P}_C$ will keep dropping as ε increases until it reaches a critical value, where the slow-down of the increasingly heavy diagonal starts to slow down the mixing.

The interesting regime is those values of $\varepsilon > 0$ where $\lambda^C_2 < \lambda^B_2$, that is, the combined chain mixes *faster* than the balanced chain. The *optimal* value of the parameter ε with respect to the bound of (2.4) is the value that minimizes λ^C_2, but the entire regimen allows an improvement in the mixing time. In general terms, if the total weight of the spectrum (that is, the sum of the absolute values of the eigenvalues) is lower, the chain mixes faster. Also the total weight first decreases as ε is slowly increased from zero, but begins to increase for larger values of epsilon. In the next section we address this and other performance issues through experimental analysis.

5. Experiments. The implementation of the combined walk depends on the information available. We suppose that the graph is stored in a format that permits queries of adjacency lists of vertices. The degree of a vertex is simply the number of elements in its adjacency list.

Should one have access to the complete information of the graph, one may apply a preprocessing step that retrieves the list of neighbors of a given vertex along with the degree to create such a database of the graph in adjacency list format. Such preprocessing takes $\mathcal{O}(m)$ time and space and could also be augmented, with no loss of asymptotic efficiency, to contain the degree of each neighbor in the adjacency lists, which would later allow higher efficiency in the implementation of the sampling method. We leave to future work a version of the sampling that estimates the vertex degrees during the walk instead of computing them by preprocessing or by lookaheads; the use of such estimates are likely to slow down the convergence of the chain to the desired distribution, as the transition probabilities will change over time.

The neighbor degrees are useful when computing the transition probabilities on the sampling side. In the absence of such precomputed neighbor degrees, which is the common case, every step of the sampler has time and space complexity $\mathcal{O}(D)$, where D is the maximum degree of the graph. Typically, the average degree of a real-world graph is much lower than the maximum degree, for which the typical cost of each step is much lower.

The computation time of the sampling as a whole, being the number of steps that are taken, depends linearly on the size of the sample needed and the mixing time of the chain. In this section we experimentally study the mixing time to provide an idea of the efficiency of the proposed method.

It is noteworthy that the above sampler construction does not require explicitly copying the vertex set: it suffices to maintain a state flag to indicate which set of transition probabilities should be used. Each of the transition probabilities needed at a vertex v, whether on the sampling side or the mixing side, is locally computable from the parameter ε, $\deg(v)$, and the degrees of the vertices in $\Gamma(v)$ (which are needed on the sampling side).

The implementation of a sampling method is follows:

1. Run the Markov chain until convergence.
2. While on the mixing side, keep running the chain.
3. Once on the sampling side, take a sample.
4. For another *independent* sample, return to step 1.
5. As an alternative to step 4, for uniformly distributed samples that are *not* independent, one may take a sample on each step where the walk is on the sampling side.

It is necessary either to determine a safe step count that should suffice for the chain to mix, or to define some alternative stopping condition to detect convergence. In our experiments, we defined the step count explicitly and took sets of independent samples, that is, remixed the chain after each sample. In doing so, we see there is no statistical difference whether the chain is started in the previously sampled vertex or in a fixed start vertex as long as the graph is connected and true convergence is reached.

5.1. Test data. We studied the behavior of the method on real-world *collaboration graphs* [41, 53, 47], a deterministic network model [19], and a closed-form graph.

A collaboration graph is a social network where the vertices represent authors of scientific articles and the edges represent the coauthorship relation. Our collaboration graphs are based on bibliography files from The Collection of Computer Science Bibliographies [1], using only the mathematical bibliographies in BibTEX format (cf. [53] for details). Different subgraphs of the collaboration graph thus compiled are used to obtain graphs of different orders.

The deterministic scale-free graph construction by Dorogovtsev, Goltsev, and Mendes (DGM) [19], based on [5], ideally suits our purposes because it is analytically easy to approach. In the model, the initial graph $G_{-1} = (V_{-1}, E_{-1})$ consists of two vertices v and w and the edge (v, w). At each discrete time step $t \geq 0$ of the process, per each $(v, w) \in E_{t-1}$, a new vertex u is added, together with edges (v, u) and (w, u). Thus at time $t = 0$, G can be represented as a triangle. See Figure 3 for an illustration of the first five generations. At time t, the number of edges is

$$|E_t| = |E_{t-1}| + 2|E_{t-1}| = 3|E_{t-1}| = 3^{t+1}, \tag{5.1}$$

as $|E_{-1}| = 1$. Similarly, the number of vertices is

$$|V_t| = |V_{t-1}| + |E_{t-1}| = 3(3^t + 1)/2. \tag{5.2}$$

Therefore the average degree of the resulting graph G_t is

$$k_t = \frac{2|E_t|}{|V_t|} = \frac{4 \cdot 3^t}{3^t + 1}. \tag{5.3}$$

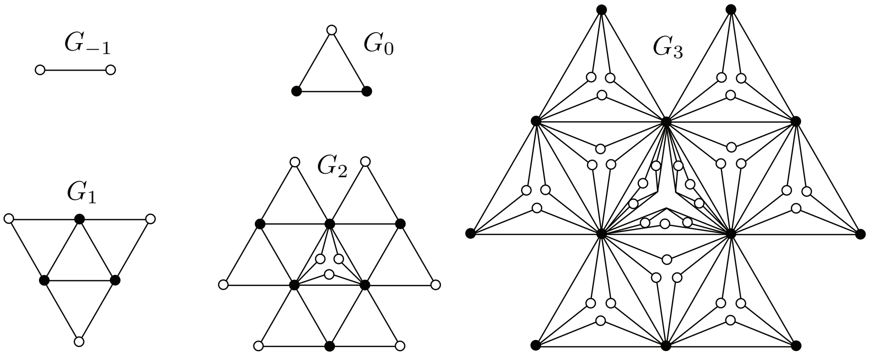

FIG. 3. *The* pseudofractal *graph (DGM) G_t for $t \in \{-1,0,1,2,3\}$ (adapted from* [19]*). Vertices added at time step t are shown in white.*

Also note that the maximum degree Δ of the graph G_t is always double the maximum degree of the previous generation and that for G_0 we have $\Delta = 2$. Hence $\Delta_t = 2^{t+1}$.

The degrees of the vertices are well behaved: the vector $\mathbf{d}$ of distinct degree values at time $t \geq 0$ is

$$\mathbf{d} = (d_1, d_2, \ldots, d_{t+1}) = (2, 2^2, 2^3, \ldots, 2^t, 2^{t+1}). \tag{5.4}$$

Letting $\eta_i = |\{v \mid v \in V_t,\ \deg(v) = d_i\}|$, the vector η of the number of vertices with degree d_i is

$$\eta = (\eta_1, \eta_2, \ldots, \eta_{t+1}) = (3^t, 3^{t-1}, 3^{t-2}, \ldots, 3^2, 3, 3). \tag{5.5}$$

For these graphs, the transition probabilities t_i that go out from a vertex that has been alive for i generations are easy to compute. The degrees are known from (5.4), giving $t_v = 2^{-(v+1)}$. Thus for the newly created vertices, the outward transition probability is $t_0 = \frac{1}{2}$. Similarly for vertices that were created on the preceding step, we have $t_1 = \frac{1}{4}$, and so forth.

We then experimented with a *closed-form* graph, that is, a graph where the neighborhood relation is implicitly defined by the vertex identifiers, which permits us to avoid the IO (input and output) overhead in the experimentation, as the graph needs not to be read from disk. One of the simplest closed-form graph constructions that can be scaled to massive orders is the *hypercube* $\mathcal{H}_b$, where the vertices are b-bit binary strings (where b is any positive integer) and two vertices are neighbors if the *Hamming distance* of the bit strings is one, that is, if the strings differ only in one bit. Figure 4 shows the construction for $b \in \{0,1,2,3\}$. The number of vertices for the hypercube is $n_{\mathcal{H}} = 2^b$ and the degree of each vertex is b, as there are exactly b positions in which the string may differ. Hence, we have $m_{\mathcal{H}} = b \cdot 2^{b-1}$ edges.

The downside is, evidently, that the hypercube is a regular graph, where uniform sampling is achieved by a regular random walk, and performing the combined walk would gain us nothing. We avoid this problem by inserting an additional hub vertex v_h with degree Δ into the hypercube, increasing n by one and and m by Δ. We used the following construction for selecting the neighbors of the hub to be able to control Δ easily and maintain the implicit definitions of the neighborhoods. Denote the b-bit hypercube by $\mathcal{H} = (V_H, E_H)$, where $V_H = \{1, 2, \ldots, 2^b\}$, and let ξ be an integer parameter to control the neighborhoods. Define

$$\Gamma(v_h) = \{v \mid (v \in V_H) \wedge (v \bmod \xi \equiv 0)\}. \tag{5.6}$$

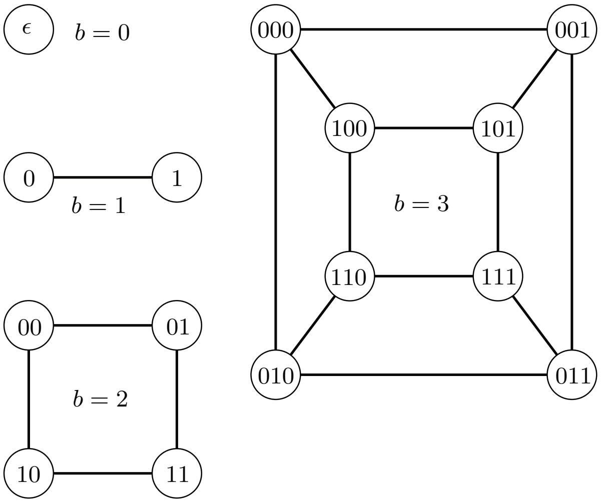

FIG. 4. *The hypercube graph for $b \in \{0, 1, 2, 3\}$. The vertex labels are shown inside the vertices; here ϵ denotes the empty string.*

Knowing ξ, we may derive[6]

$$\Delta = \left\lfloor \tfrac{1}{\xi} 2^b \right\rfloor . \tag{5.7}$$

Note that a larger value of ξ gives a smaller-degree hub. We denote the augmented hypercube by $\mathcal{H}_\xi$.

5.2. Mixing time and coverage. We computed the eigenvalue spectra of the regular chain's transition matrix and the balanced chain's transition matrix for a few generations of the DGM model; the plots are shown in Figure 5. It is evident from the plots that the eigenvalues of the balanced chain are larger than those of the regular walk, which gives evidence of slow mixing, at least partially caused by the self-loop probabilities.

In Figure 6, ε is varied from zero to one for the same four DGM graphs. On the left, the magnitude of $|\lambda_2^C|$ is shown, and on the right, we plot

$$W = \frac{1}{n} \sum_{i=1}^{n} |\lambda_i^C| \tag{5.8}$$

as another estimate of the mixing time. This normalized sum is closely related to that of the *energy* of the graph [30], in which the spectrum of the adjacency matrix is used instead of the spectrum of the a transition matrix, and is even more closely related so to the *Laplacian energy* [31]—for more information on the relationship of the spectrum of the Laplacian of a graph to that of the transition matrix of a regular random walk, see [44].

The horizontal lines in Figure 6 correspond to the magnitude of the second eigenvalue and to the spectral weight of the regular and the balanced random walks. Those of the regular walks (that mix fast) are always clearly below the values corresponding to the balanced walks, and are also below all those attained by the combined walk,

[6] The derivation is straightforward and left to the reader.

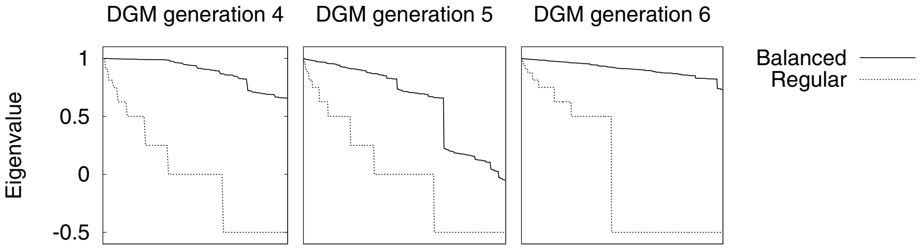

FIG. 5. *The spectra of three generations of the DGM model for two different transition matrices—that of the regular chain (equation* (4.1)*) and that of the balanced chain (equation* (4.4)*).*

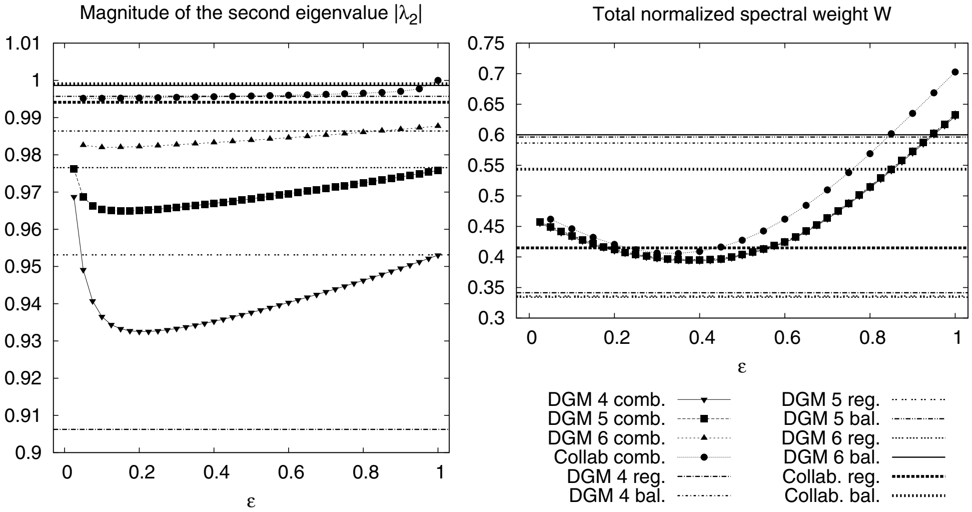

FIG. 6. *Left: The magnitude of the second eigenvalue $|\lambda_2|$ of the transition matrices of the three types of Markov chains. Right: The normalized total spectral weight of the transition matrices. The horizontal lines are the values of the regular and balanced walks, and the curves are those of the combined walk, where the behavior depends on the value of the parameter ε. All plots include a* 503*-vertex collaboration graph and three generations of the DGM model and were computed with Octave.*

as can be expected. The gain of the combined walk when compared to the balanced walk is, however, significant. For the collaboration graph (heavy dotted lines), the difference between the regular and the balanced walk is smaller than for the DGM graphs.

Higher generations of the DGM model appear to mix worse than preceding generations in light of (2.4), as the value of the second eigenvalue grows with the generation as follows: 0.9567 for the third generation, 0.9864 for the fourth, 0.9957 for the fifth, 0.9986 for the sixth, and 0.9996 for the seventh (computed on Octave [21] using `eigs` from ARPACK [39]). In terms of the normalized spectral weight there is very little difference. Also, an interesting effect is that while the spectral weight of the regular chain on the DGM model decreases in higher generations, for the balanced walk it *increases*, as the balanced walk gets stuck due to the increasing difference between the average and the maximum degree, as both the number of nodes with two neighbors and the degree of the three hubs increase rapidly. The curves for the combined walk practically overlap for these three generations.

It is notable that a wide range of values of ε offer an improvement in these spectral measures of the mixing time when compared with the balanced walk—the curves cross the horizontal line rather late. Judging only by the second eigenvalue, almost all values of ε are good: the range [0.1, 0.9] improves upon the mixing time of the balanced chain with the best values in $[0.1, 0.2]$. In terms of the normalized spectral weight, $\varepsilon < 0.8$ is always an improvement, with the optimal value being in the interval [0.3, 0.5]. An interesting observation is that the optimal values of ε are neither the same nor similar for these two estimates of the mixing time.

The general conclusion from Figure 6 is that the selection of a good value of ε is not hard, although it could be improved by the introduction of some heuristics. We leave this aspect to future work.

For a simulation-based estimate of the mixing time, we must have a way to detect when the chain has mixed. Examining whether a Markov chain has converged to the stationary distribution can be done, for example, by measuring the total variation distance $\mathcal{D}$ of (2.3).

An experimental evaluation of the total variation distance for a Markov chain can be done by running several instances of the chain from the same start position and calculating an estimate based on the state distribution over the independent instances [9]. Denoting the number of instances run by I and the number of instances that are at state v at time t by $f_t(v)$, a conservative estimate that slightly overestimates the total variation distance at time step t is

$$\mathcal{D}_{\text{est}} = 1 - \sum_{f_t(v)\neq 0} \min\left\{\frac{f_t(v)}{I}, \frac{1}{n}\right\}. \tag{5.9}$$

The bias of the estimator can be analyzed for different stationary distributions. For example, take I instances over a chain with n states that has the uniform distribution as the stationary distribution. Assume that at time t_m the instances have mixed, and hence the probability of finding any single instance j in state $v \in V$ is $p = \frac{1}{n}$. The number of instances $f_t(v)$ in a state v at time t is binomially distributed with $p = \frac{1}{n}$. As the estimate takes the minimum of $\frac{1}{n}$ and the fraction $\frac{f_t(v)}{I}$, we need only consider states in which there are *less* than $\lceil\frac{1}{n}\rceil$ instances, as these states are the ones that introduce bias to the estimate. The *deficit* of a vertex with frequency $f_t(v) < \lceil\frac{1}{n}\rceil$ is

$$\frac{1}{n} - \frac{f_t(v)}{I}. \tag{5.10}$$

Combining the probabilities that there were exactly $j \in [0, \lfloor\frac{I}{n}\rfloor]$ instances at each of the n states and the corresponding deficits, the total bias is

$$\mathcal{B} = n \sum_{j=0}^{\lfloor\frac{I}{n}\rfloor} p^j (1-p)^{\lfloor\frac{I}{n}\rfloor - j} \binom{I}{j} \left(\frac{1}{n} - \frac{j}{I}\right). \tag{5.11}$$

Figure 7 shows the bias estimate of (5.11) for four different values of n, assuming a uniform stationary distribution and a fully mixed chain.

We studied the convergence of the sampling methods to their respective stationary distributions over the vertex set by estimating the total variation distance between

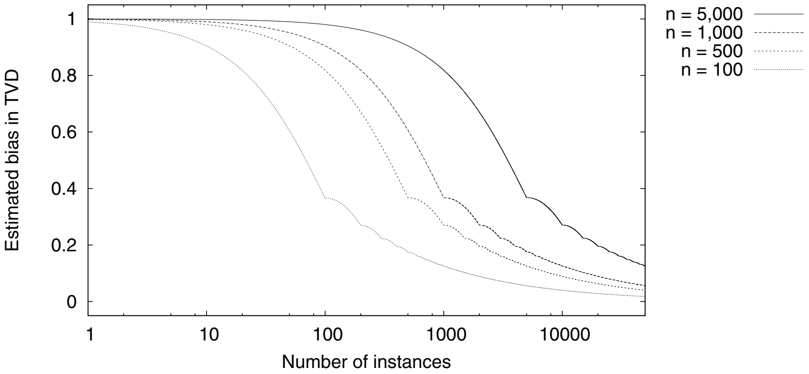

Fig. 7. *The estimated bias of* (5.11) *to the estimator of total variation distance* (5.9) *for four different values of* $n = |V|$. *Note that when the number of instances is a multiple of the state count, the curve displays a knee bend, as the possibility of dividing the instances evenly over the state set decreases the total bias.*

the obtained and the stationary distribution with (5.9). For the combined walk, *only* those instances that were currently in a sampling state (instead of being on the mixing side of the construction) were included in the estimate, and hence the estimates for the other two walks are based on a greater number of independent instances than those of the combined walk. Using such estimation, the stationary distribution to which the combined walk should converge is the uniform distribution. Figure 8 shows the estimate for the DGM construction, generations five and seven, and for two collaboration graphs based on natural data.

As the bias of the estimate depends on the number of instances for which it is calculated, we have plotted in Figure 9 the actual number of instances on the sampling side of the combined walk for the data of Figure 8.

We studied the tendency of the balanced chain to unwanted locality by plotting the percentage of vertices that have received at least one visit during a long random walk; we call this percentage *coverage*. The regular chain will visit each vertex at least once much more rapidly than the balanced chain, as the balanced chain stalls in local neighborhoods—this can be seen in Figures 10 and 11.

5.3. Sample quality and parameter estimation. We evaluated the quality of the samples obtained by comparing the degree distribution of samples obtained by the different methods to the real degree distribution. From Figure 12 it can be seen that the regular random walk samples vertices preferentially to their degree, whereas the other methods (that weigh vertices inversely to their degree) maintain an indifference to vertex degree and obtain samples that preserve the form of the original degree distribution.

We also studied the accuracy of the estimate of k that one gets by keeping track of the number of visits on the sample side and the total accumulated degree. We ran 30 independent 10,000-step combined walks with initial vertices chosen uniformly at random. We computed such an estimate on the average degree, shown in Figure 13. Almost from the start, the estimate stays near the true value, permitting on-the-fly evaluation of the value of α, and hence knowing when the chain should be mixed.

5.4. Performance and scalability. We wanted to estimate the performance and scalability of the method and chose to experiment first on the closed-form graph

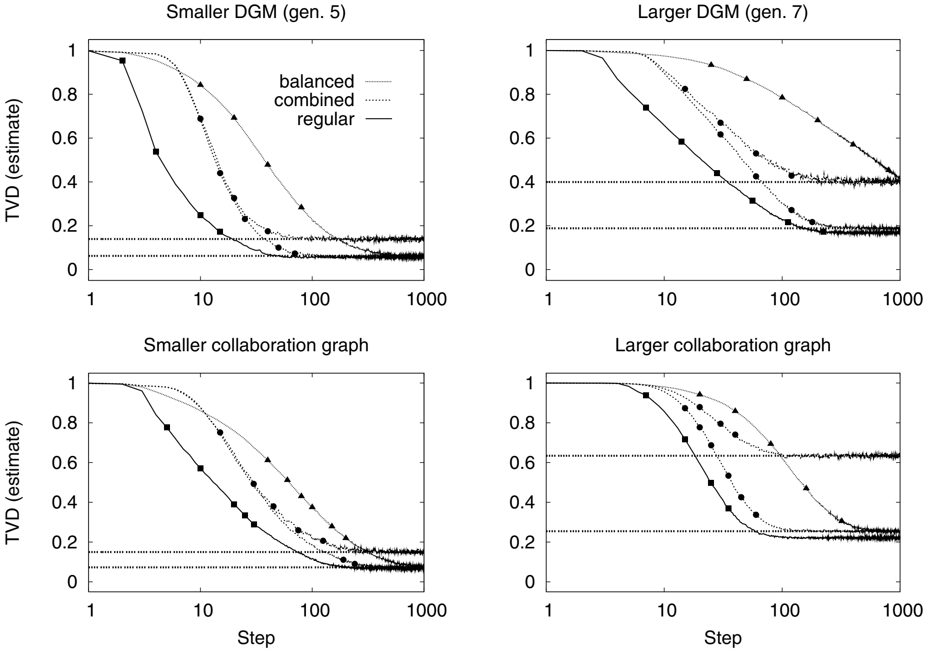

FIG. 8. *Values of the estimated $\mathcal{D}$ cover time for the balanced* (▲, *equation* (4.4)), *combined* (●, *equations of section* 4.3, *two curves as explained below*), *and regular* (■, *equation* (4.1)) *chains. The bias of the uniform distribution estimate* (5.11) *for each graph is shown as the lower dotted line. The estimate is calculated over a set of $I = 15,000$ independent walks in two DGMs, and two collaboration graphs. Note that for the combined chain ($k = 4, \varepsilon = 0.25$), the expected number of instances on the sampling side of the combined walk is $\alpha I \leq 15,000$, and hence the bias is larger (the upper horizontal line on the plot). Hence, for ease of comparison, we also ran the combined walk for $I' = \alpha^{-1} I$ to achieve the same expected bias (drawn as a lower horizontal line) as a balanced walk. All walks were started at a fixed vertex, initially chosen at random. The smaller (fifth-generation) DGM has $n = 366$ and $m = 729$, and the larger (seventh-generation) DGM has $n = 3,282$ and $m = 6,561$. The smaller collaboration graph has $n = 503$ and $m = 828$, and the larger collaboration graph has $n = 5,909$ and $m = 13,510$.*

$\mathcal{H}_\xi$ of section 5.1 to perform experiments on larger graphs. We used different values of ξ to create a few different degree distributions. For the rest of this section, for simplicity, we denote this hub-augmented graph $\mathcal{H}_\rho$ by $G = (V, E)$, its order by n, and its size by m. We fixed the bit-string length to $b = 30$, which gave us a graph with $n = 1,073,741,825$ and $m = 16,106,127,360 + \Delta$, where Δ is given by (5.7). The resulting hub degrees when $b = 30$ are shown in Table 2 for some values of ξ.

We first took a half-million independent samples using an approximately 200,000-step walk per sample; if the walk was on the mixing side on step number 200,000, we took the sample as it first hit the sampling side. This experiment was performed for $\varepsilon \in \{0.2, 0.4, 0.6, 0.8\}$. We kept a record of some measures of interest, the most relevant of which are shown in Table 3. For comparison, we also ran the same experiments on the regular random walk.

By definition, smaller values of ε yield fewer mode switches and fewer self-loops, whereas the growth in the extent to which the graph is traversed causes an increase in the number of lookaheads needed to compute the neighbor degrees, and hence the running time of the walk increases, as seen in Table 3. Naturally, ξ has no significant effect on the number of mode switches made nor on the number of self-loops followed,

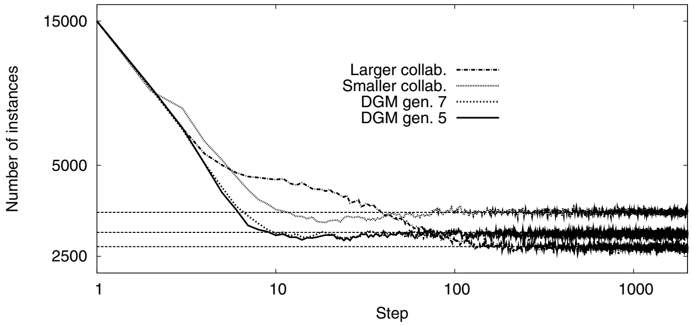

Fig. 9. *The number of walk instances out of the total of $I = 15,000$ instances that are on the sampling side of the combined walk at each step; the data used is the same as in Figure* 8. *We use $\varepsilon = 0.4$; this gives $\alpha \approx 0.20$ for DGM generations* 5 *and* 7, *$\alpha \approx 0.23$ for the smaller collaboration graph, and $\alpha \approx 0.18$ for the larger. The theoretical value to which the plots are expected to converge, $I \cdot \alpha$, is shown in the plot as a vertical line for each value of α. The graphs are the same as those used in Figure* 8.

Table 2

The hub degree Δ and the edge count for the augmented hypergraph with different values of ξ for $b = 30$; there is always exactly one hub by construction. The immediate neighbors of the hub are called router vertices and have degree $b+1$, and the remaining unaffected vertices are called normal and have degree b. The average degree (≈ 30) hardly changes as the number of edges m is so high in comparison to the number of vertices, $n = 1,073,741,825$, and ξ has only a small effect on m with the value chosen for b. Similarly, $\alpha \approx 0.03226$ for all values of ξ used.

ξ	Δ	m
30^2	1,193,046	16,107,320,421
30^3	39,768	16,106,167,143
30^4	1,325	16,106,128,700
30^5	44	16,106,127,419

Table 3

Measures of interest in the augmented hypercube experiments for $\xi = 30^4$ and four values of ε. All walks were of 200,000 *steps. The columns indicate the number of samples taken per second of runtime (SPS—for the regular random walk this was* 82.621, *which is naturally higher, as less computation is required per sample), the number of mode switches (MS) per sample, the number of self-loops (SL) followed per sample, the number of steps actually taken (SC), and the effective step count (ESC) computed by subtracting from the number of steps taken the number of self-loops followed, all of these being averages over the samples taken, rounded to the nearest integer.*

ε	SPS	MS	SL	SC	ESC
0.2	32.236	2,582	36,152	200,145	163,993
0.4	35.589	5,162	72,285	200,072	127,788
0.6	40.914	7,743	108,414	200,048	91,634
0.8	45.160	10,324	144,544	200,036	55,493

as these measures both depend on ε. Also note that k is nearly unaffected by the values of ξ used in this experiment, for which only the measures for $\xi = 30^4$ are included in Table 3, although the experiment was run also for $\xi = 30^2$ and $\xi = 30^3$.

A noteworthy point is that the self-loops can be avoided during the algorithm execution: instead of following a self-loop, the number of steps spent in self-loops can be generated at random using the geometric distribution. Leaving a vertex is modeled as a success in a Bernoulli trial, whereas following a self-loop is a failure, as mentioned in section 3 for the coin-flip random walk. The number of effective steps

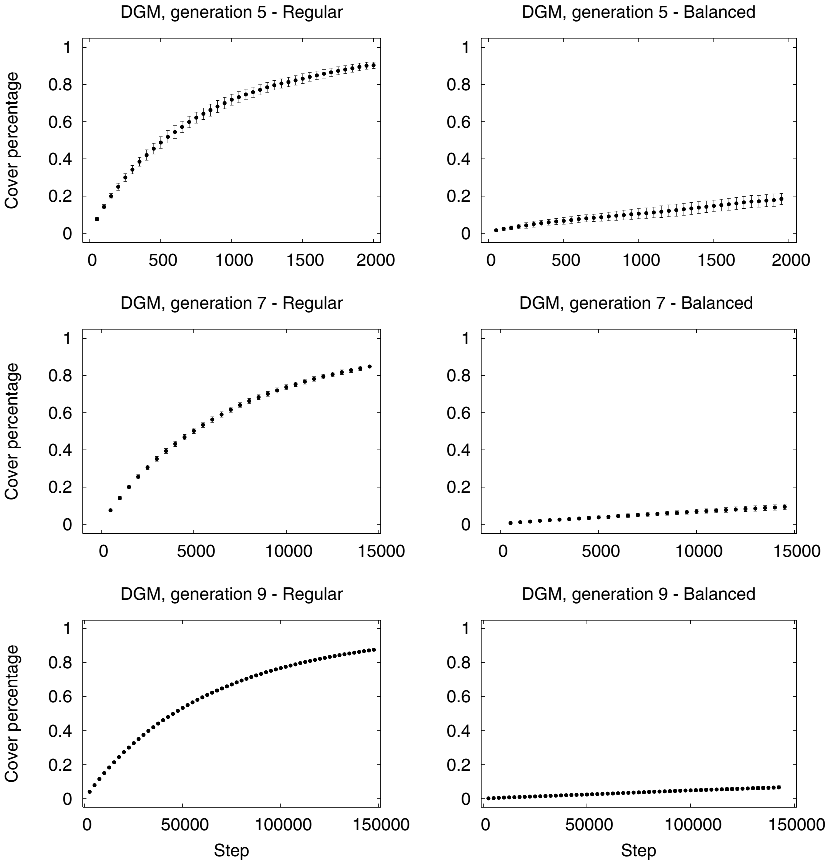

FIG. 10. *The coverage achieved by the regular (left) and the balanced (right) walks at each step. In all six plots, averages and standard deviations of* 50 *independent walks are shown.*

taken to actually traverse the graph depends on ε, and with $\varepsilon = 0.8$ we see that in fact only a quarter of the transitions actually explore the graph. This is important to take into account when defining the step count: for smaller values of ε, smaller step counts suffice to explore the graph more widely.

We studied in particular the visit counts and sampling frequencies in a small experiment of taking one million samples of the augmented hypercube using $\rho = 3$ with 20,000-step and 100,000-step walks using a combined walk with $\varepsilon = 0.5$ and a regular walk. We compared the sampling frequencies to the analytical *expected* sample counts, based on the degree distribution of the input graph and the sample count assuming uniformity. Intuitively, the combined walk visits and samples the higher-degree vertices less than the regular walk.

We ran an experiment to study the number of distinct vertices visited by the chain and examined as lookahead when obtaining a single sample, varying the length of the walk. It is important to note that the number of steps needed for mixing depends heavily on the graph structure, and the number of steps necessarily affects the number of distinct vertices visited. We ran walks of different lengths to obtain a single sample each from the hypercube construction. The results are shown in Figure 14 in terms of steps taken, unique vertices visited, lookaheads required, and total

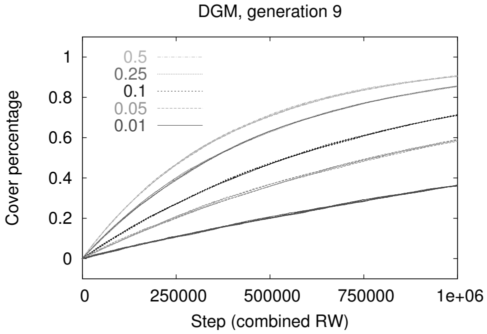

FIG. 11. *The behavior of the combined random walk on a ninth-generation graph for different values of ε. Increase in coverage is measured only at sampling steps; i.e., the vertex visits of the mixing side of the construction are not counted in the coverage but are present in the step count. As expected, the topmost curve is the fastest coverage achieved with $\varepsilon = 0.5$ and the slowest coverage resulting from $\varepsilon = 0.01$.*

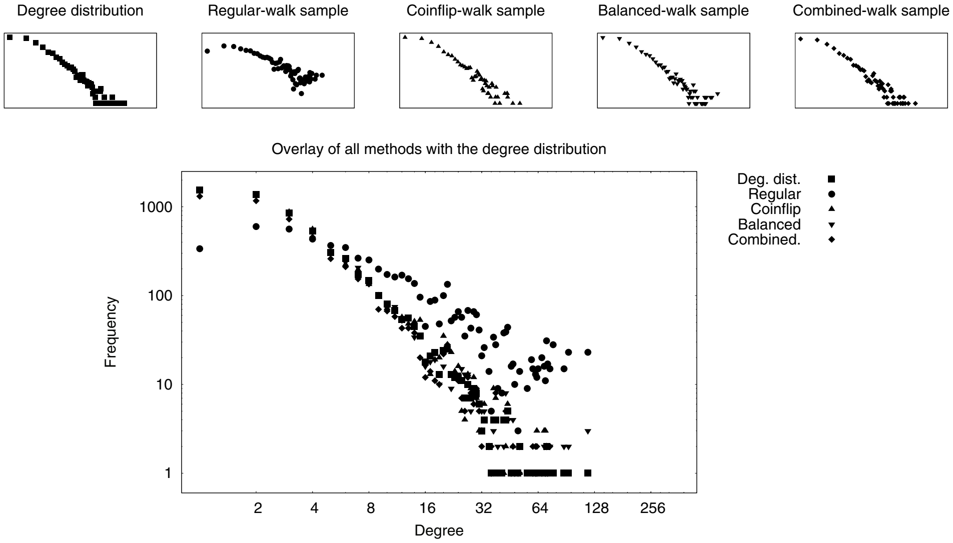

FIG. 12. *Sampling frequency of vertices by their degree from a collaboration graph with $n = 5,909$ and $m = 13,510$. A set of n independent samples was taken, fully restarting from the same randomly chosen vertex, allowing the walks to mix for $t \geq 5,000$. The small plots show the actual degree distribution and the frequency distributions for all walks, and the larger plot overlays all these distributions. We used $\varepsilon = 0.25$ for the combined walk. Note that both axes have logarithmic scale.*

coverage and steps per unique vertex. The plots show that, on average, a previously unseen vertex is visited every two steps taken by the chain, and the frequency of discovering new vertices (unsurprisingly) decreases a little for longer walks. Also, even with the lookaheads, the proposed method achieves locality, as only a small fraction of vertices need to be examined as lookaheads.

The samples taken in these walks were also analyzed to show how their distribution changes as more steps are taken per sample. Figure 15 shows 20-bin histograms over the 5,000 samples taken, grouping the vertices per their label (zero is the hub vertex v_h and its additional neighbors are evenly distributed by label as defined by (5.6)).

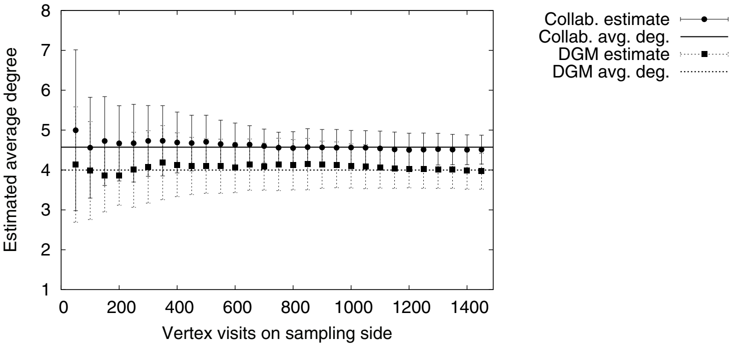

FIG. 13. *The average and standard deviation of the estimate of the average degree based on vertices visited on the sampling side of a combined random walk. The average is over* 30 *independent walks. The actual average degree of the graphs is shown with straight lines—the graphs are the larger collaboration graph and the seventh-generation DGM that were used in Figure* 8.

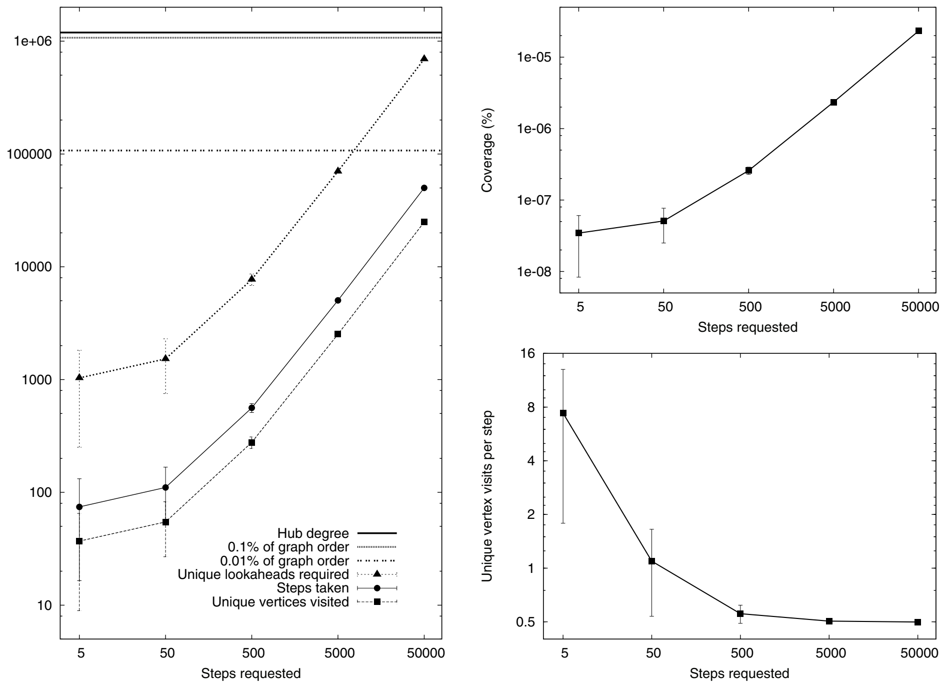

FIG. 14. *The scaling behavior with respect to the number of previously unseen vertices visited by a walk and examined as lookahead in the combined walk on the hypercube construction with* $b = 30$, $\xi = 30^3$, *and* $\varepsilon = 0.5$. *Average and standard deviation over* 30 *replicas for five walk lengths:* 5, 50, 500, 5,000, *and* 50,000 *steps. All walks were restarted at a fixed vertex and took a single sample.*

We also recorded the number of unique vertices visited by the chain in the 13th-generation DGM graph, with n = 2,391,486 and m = 4,782,969. We obtained a sample of 5,000 vertices, restarting the chain for each sample taken at a fixed vertex, using 100-step walks and $\varepsilon = 0.1$. During the sampling, we recorded the complete list of vertices visited and then calculated how many distinct vertices were in fact visited on these 5,000 walks. This was repeated over 30 replicas and then the results

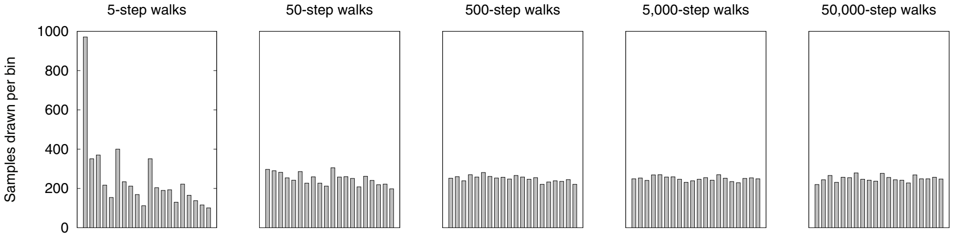

FIG. 15. *20-bin histograms of the samples drawn, grouped by vertex label, in walks of different lengths (analyzed in Figure* 14*) on the hypercube construction. Note that the distribution is still far from uniform when using* 5*-step walks but is much closer to uniform using any other step count from* 50 *to* 50, 000 *steps.*

were averaged. The number of unique vertices visited was on average 179,361, that is, 7.5 percent of the graph. This, dividing by 5,000, gives 35 unique visits per sample. However, if only a single sample is taken, the advantage of redundancy in vertex examinations over the distinct samples is lost. Hence, for comparison, we ran 30 replicas taking a single sample. For single samples, using the same parameters, the number of unique vertices visited was on average approximately 64, which is 0.003 percent of the graph.

We repeated this same experiment, with the same parameters, on a collaboration graph of 108,624 vertices and 333,546 edges. The average of unique vertex visits over 30 replicas was 65,834 for the 5,000 samples in total—approximately 60 percent of the graph—giving 13 unique visits per sample. For a single sample, the number of unique vertex visits was on average approximately 71, which is 0.01 percent of the graph.

Changes in the parameter ε or the length of the walk would naturally affect the number of unique vertices visited. It is clear from these results that also the structure of the graph plays an important role in the extent to which the graph is traversed, as it also affects the length of the walk required for reaching the stationary distribution, as the collaboration graph and the DGM 13th-generation graph yield distinct results. We leave to further work the auto-adaptation of the parameter ϵ based on structural information gathered during the execution of the sampling method.

6. Conclusions. In this paper, we presented a construction for sampling undirected graphs uniformly by local computation. Two classic Markov chains are combined into one: a regular random walk provides rapid mixing to the construction, whereas a balanced walk permits sampling from the uniform distribution. We showed formally the correctness of the construction and verified by experiments on natural and artificial graphs that it has the desired properties. We look forward to constructing sampling methods for directed and weighted graphs. The asymmetry caused by weights or edge directions breaks the validity of the presented construction.

As future work, it would also be useful to study how well different kinds of sampling methods preserve different structural properties of large (natural or generated) networks. Chakrabarti et al. [10] propose the *NetMine* tool for analyzing large graphs by providing views on different structural properties of graph instances given as input. Among other features, the tool aims to identify vertices or subgraphs that are structural outliers, i.e., pointing out nonuniformities in the network structure. The scalability issues in the implementation of such tools could be avoided by first employing efficient sampling mechanisms to obtain a preview of the plots that are assumed to be of interest, and then selecting the measures to be calculated for the full data set based on observations made on the preview plots.

Acknowledgments. The author thanks Harri Haanpää for useful conversations and the anonymous reviewers for their useful observations that helped to improve the presentation of this work.

REFERENCES

[1] A.-C. Achilles, *The Collection of Computer Science Bibliographies*, http://liinwww.ira.uka.de/bibliography/ (2002).

[2] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, *On the bias of traceroute sampling, or: Power-law degree distributions in regular graphs)*, in Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing (STOC), H. N. Gabow and R. Fagin, eds., ACM Press, New York, 2005, pp. 694–703.

[3] P. Baldi, P. Frasconi, and P. Smyth, *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*, John Wiley & Sons, Chichester, UK, 2003.

[4] Z. Bar-Yossef, S. R. Kumar, and D. Sivakumar, *Sampling algorithms: Lower bounds and applications*, in Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing (STOC), ACM Press, New York, 2001, pp. 266–275.

[5] A.-L. Barabási, E. Ravasz, and T. Vicsek, *Deterministic scale-free networks*, Phys. A, 299 (2001), pp. 559–564.

[6] E. Behrends, *Introduction to Markov Chains, with Special Emphasis on Rapid Mixing*, Vieweg & Sohn, Braunschweig/Wiesbaden, Germany, 2000.

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

[8] A. Z. Broder, S. R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, *Graph structure in the Web*, Computer Networks, 33 (2000), pp. 309–320.

[9] S. P. Brooks, P. Dellaportas, and G. O. Roberts, *An approach to diagnosing total variation convergence of MCMC algorithms*, J. Comput. Graph. Statist., 6 (1997), pp. 251–265.

[10] D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos, and G. Blelloch, *Netmine: New mining tools for large graphs*, presented at the 4th SIAM International Conference on Data Mining, Workshop on Link Analysis, Counterterrorism, and Privacy, Buena Vista, FL, 2004. Available online at http://www-users.cs.umn.edu/∼aleks/sdm04∼/talks.htm.

[11] S. Chib and I. Jeliazkov, *Accept-reject Metropolis-Hastings sampling and marginal likelihood estimation*, Statist. Neerlandica, 59 (2005), pp. 30–44.

[12] K. L. Clarkson, *Further applications of random sampling to computational geometry*, in Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC), ACM Press, New York, 1986, pp. 414–423.

[13] K. L. Clarkson, *Applications of random sampling in computational geometry*, II, in Proceedings of the Fourth Annual Symposium on Computational Geometry, ACM Press, New York, 1998, pp. 1–11.

[14] A. Clauset and C. Moore, *Accuracy and scaling phenomena in Internet mapping*, Phys. Rev. Lett., 94 (2005), article 018701.

[15] F. Comellas and S. Gago Álvarez, *Spectral bounds for the betweenness of a graph*, Linear Algebra Appl., 423 (2007), pp. 74–80.

[16] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani, *Exploring networks with traceroute-like probes: Theory and simulations*, Theoret. Comput. Sci., 355 (2006), pp. 6–24.

[17] S. Datta and H. Kargupta, *Uniform data sampling from a peer-to-peer network*, in Proceedings of the Twenty-Seventh International Conference on Distributed Computing Systems, IEEE Computer Society, Washington, DC, 2007, p. 50.

[18] N. Deo and P. Gupta, *Sampling the Web graph with random walks*, Congr. Numer., 149 (2001), pp. 65–73.

[19] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, *Pseudofractal scale-free web*, Phys. Rev. E, 65 (2002), article 066122.

[20] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of networks*, Adv. Phys., 51 (2002), pp. 1079–1187.

[21] J. W. Eaton et al., *GNU Octave—A High-Level Language for Numerical Computations*, http://www.gnu.org/software/octave/ (2010).

[22] D. Eppstein and J. Wang, *Fast approximation of centrality*, J. Graph Algorithms Appl., 8 (2004), pp. 39–45.

[23] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power-law relationships of the Internet topology*, in Proceedings of the ACM SIGCOMM'99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM Press, New York, 1999, pp. 251–262.

[24] N. G. Feamster, *Proactive Techniques for Correct and Predictable Internet Routing*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2006.

[25] S. Floyd and E. Kohler, *Internet research needs better models*, ACM SIGCOMM Comput. Comm. Rev., 33 (2003), pp. 29–34.

[26] B. Gärtner and E. Welzl, *Random sampling in geometric optimization: New insights and applications*, in Proceedings of the Sixteenth Annual Symposium on Computational Geometry, ACM Press, New York, 2000, pp. 91–99.

[27] D. Gillman, *A Chernoff bound for random walks on expander graphs*, in Proceedings of the Thirty-Fourth Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Washington, DC, 1993, pp. 680–691.

[28] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 8271–8276.

[29] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 3rd ed., Oxford University Press, Oxford, UK, 2001.

[30] I. Gutman, *The energy of a graph: Old and new results*, in Algebraic Combinatorics and Applications, A. Betten, A. Kohnert, R. Laue, and A. Wassermann, eds., Springer-Verlag, Berlin, 2001, pp. 196–211.

[31] I. Gutman and B. Zhou, *Laplacian energy of a graph*, Linear Algebra Appl., 414 (2006), pp. 29–37.

[32] Y. Ji, X. Xu, and G. D. Stormo, *A graph theoretical approach to predict common RNA secondary structure motifs including pseudoknots in unaligned sequences*, Bioinformatics, 20 (2004), pp. 1591–1602.

[33] N. Kahale, *A semidefinite bound for mixing rates of Markov chains*, Random Structures Algorithms, 11 (1998), pp. 299–313.

[34] D. R. Karger, *Random sampling in cut, flow, and network design problems*, in Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing (STOC), ACM Press, New York, 1994, pp. 648–657.

[35] D. R. Karger, *Random Sampling in Graph Optimization Problems*, Ph.D. thesis, Stanford University, Stanford, CA, 1995.

[36] J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, *The Web as a graph: Measurements, models, and methods*, in Proceedings of the Fifth Annual International Conference on Computing and Combinatorics, T. Asano, H. Imai, D. Lee, S. Nakano, and T. Tokuyama, eds., Lecture Notes in Comput. Sci. 1627, Springer, Berlin, 1999, pp. 1–17.

[37] D. Krioukov, K. C. Claffy, K. Fall, and A. Brady, *On compact routing for the internet*, ACM SIGCOMM Comput. Comm. Rev., 37 (2007), pp. 41–52.

[38] V. Krishnamurty, J. Sun, M. Faloutsos, and S. Tauro, *Sampling Internet topologies: How small can we go?*, in Proceedings of the International Conference on Internet Computing (Las Vegas, NV, 2003,) H. R. Arabnia and Y. Mun, eds., CSREA Press, Athens, GA, pp. 577–580.

[39] R. Lehoucq, K. Maschhoff, D. Sorensen, and C. Yang, Arpack—*Arnoldi Package*, http://www.caam.rice.edu/software/ARPACK/ (2008).

[40] M. Mihail, A. Saberi, and P. Tetali, *Random walks with lookahead on power law random graphs*, Internet Math., 3 (2006), pp. 147–152.

[41] M. E. J. Newman, *The structure of scientific collaboration networks*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 404–409.

[42] M. E. J. Newman, *A Measure of Betweenness Centrality Based on Random Walks*, preprint, http://arxiv.org/abs/cond-mat/0309045 (2003).

[43] P. Orponen and S. E. Schaeffer, *Efficient Algorithms for Sampling and Clustering of Large Nonuniform Networks*, preprint, http://arxiv.org/cond-mat/0406048 (2004).

[44] P. Orponen, S. E. Schaeffer, and V. A. Gaytán, *Locally Computable Approximations for Spectral Clustering and Absorption Times of Random Walks*, preprint, http://arxiv.org/abs/0810.4061 (2008).

[45] V. Paxson, *End-to-end routing behavior in the internet*, ACM SIGCOMM Comput. Comm. Rev., 36 (2006), pp. 41–56.

[46] V. Paxson and S. Floyd, *Why we don't know how to simulate the Internet*, in Proceedings of the 1997 Winter Simulation Conference, ACM Press, New York, 1997, pp. 1037–1044.

[47] S. E. Schaeffer, *Algorithms for Nonuniform Networks*, Research Report A102, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, 2006.

[48] B. Selman, H. A. Kautz, and B. Cohen, *Local search strategies for satisfiability testing*, in Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge, D. S. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 521–532.

[49] A. Sinclair, *Improved bounds for mixing rates of marked chains and multicommodity flow*, in Proceedings of the First Latin American Symposium on Theoretical Informatics, Lecture Notes in Comput. Sci. 583, Springer-Verlag, London, 1992, pp. 474–487.

[50] A. Sinclair, *Algorithms for Random Generation & Counting: A Markov Chain Approach*, Birkhäuser Boston, Boston, MA, 1993.

[51] A. Tanay, R. Sharan, and R. Shamir, *Discovering statistically significant biclusters in gene expression data*, Bioinformatics, 18 (2002), pp. S136–S144.

[52] L. Tierney, *Markov chains for exploring posterior distributions*, Ann. Statist., 22 (1994), pp. 1701–1762.

[53] S. E. Virtanen, *Properties of Nonuniform Random Graph Models*, Research Report A77, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, 2003.

[54] D. Vukadinović, P. Huang, and T. Erlebach, *On the spectrum and structure of Internet topology graphs*, in Proceedings of the Second International Workshop on Innovative Internet Computing Systems, H. Unger, T. Böhme, and A. R. Mikler, eds., Lecture Notes in Comput. Sci. 2346, Springer-Verlag GmbH, Berlin/Heidelberg, 2002, pp. 83–95.

[55] W. Wei, J. Erenrich, and B. Selman, *Towards efficient sampling: Exploiting random walk strategies*, in Proceedings of the Nineteenth National Conference on Artificial Intelligence and Sixteenth Conference on Innovative Applications of Artificial Intelligence, D. L. McGuinness and G. Ferguson, eds., AAAI Press/The MIT Press, Menlo Park, CA/Cambridge, MA, 2004, pp. 670–676.

[56] Y. Xu, V. Olman, and D. Xu, *Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees*, Bioinformatics, 18 (2002), pp. 536–545.