Random Walks on Random Graphs

Colin Cooper[1] and Alan Frieze[2⋆]

[1] Department of Computer Science, King's College, University of London, London WC2R 2LS, UK
[2] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213, USA.

1. Introduction

The aim of this article is to discuss some of the notions and applications of random walks on finite graphs, especially as they apply to random graphs. In this section we give some basic definitions, in Section 2 we review applications of random walks in computer science, and in Section 3 we focus on walks in random graphs.

Given a graph $G = (V, E)$, let $d_G(v)$ denote the degree of vertex v for all $v \in V$. The random walk $\mathcal{W}_v = (\mathcal{W}_v(t), t = 0, 1, \ldots)$ is defined as follows: $\mathcal{W}_v(0) = v$ and given $x = \mathcal{W}_v(t)$, $\mathcal{W}_v(t+1)$ is a randomly chosen neighbour of x.

When one thinks of a random walk, one often thinks of Polya's Classical result for a walk on the d-dimensional lattice Z^d, $d \geq 1$. In this graph two vertices $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$ are adjacent iff there is an index i such that (i) $x_j = y_j$ for $j \neq i$ and (ii) $|x_i - y_i| = 1$. Polya [33] showed that if $d \leq 2$ then a walk starting at the origin returns to the origin with probability 1 and that if $d \geq 3$ then it returns with probability $p(d) < 1$. See also Doyle and Snell [22].

A random walk on a graph G defines a Markov chain on the vertices V. If G is a finite, connected and non-bipartite graph, then this chain has a stationary distribution π given by $\pi_v = d_G(v)/(2|E|)$. Thus if $P_v^{(t)}(w) = \mathbf{Pr}(\mathcal{W}_v(t) = w)$, then $\lim_{t\to\infty} P_v^{(t)}(w) = \pi_w$, independent of the starting vertex v.

In this paper we only consider finite graphs, and we will focus on two aspects of a random walk: The *Mixing Time* and the *Cover Time*.

1.1. Mixing Time For $\epsilon > 0$ let

$$T_G(\epsilon) = \max_v \min\left\{t : \ ||P_v^{(t)} - \pi||_{TV} \leq \epsilon\right\},$$

where

$$||P_v^{(t)} - \pi||_{TV} = \frac{1}{2}\sum_w |P_v^{(t)}(w) - \pi_w|$$

is the Total Variation distance between $P_v^{(t)}$ and π.

We say that a random walk on G is *rapidly mixing* if $T_G(1/4)$ is $poly(\ln|V|)$, where $\ln = \log_e$ is the natural logarithm. The choice of $1/4$ is somewhat arbitrary, any constant strictly less than $1/2$ will suffice. Rapidly mixing Markov chains are extremely useful and we will have more to say on them in Sections 2.1.1 – 2.1.3.

1.2. Cover Time For $v \in V$ let C_v be the expected time taken for a simple random walk W on G starting at v, to visit every vertex of G. The *vertex cover time* C_G of G is defined as $C_G = \max_{v\in V} C_v$. The (vertex) cover time of connected graphs has been extensively studied. It is a classic result of Aleliunas, Karp, Lipton, Lovász and Rackoff [4] that $C_G \leq 2m(n-1)$. It was shown by Feige [24], [25], that for any connected graph G, the cover time satisfies $(1-o(1))n\ln n \leq C_G \leq (1+o(1))\frac{4}{27}n^3$. As an example of a graph achieving the lower bound, the complete graph K_n has cover time determined by the Coupon Collector problem. The *lollipop* graph consisting of a path of length $n/3$ joined to a clique of size $2n/3$ gives the asymptotic upper bound for the cover time. We will have more to say on the cover time in Sections 2.2 and 3.

2. Applications in Computer Science:

2.1. Rapid Mixing

2.1.1. Sampling and Counting Let Δ denote the maximum degree of a graph $G = (V, E)$. Suppose

⋆ Supported in part by NSF grant CCF0502793.

now that we wish to find a *uniform random* proper colouring of G using $k \geq \Delta + 1$ colours. By proper we mean that adjacent vertices get different colours. We can easily generate one such colouring via a greedy algorithm, but it will certainly not be a random colouring. The distribution of random colourings of G is complex. The only known approach to this sampling problem is via random walk. Let Ω denote the set of all proper k colourings of G. This will most likely be of exponential size in $n = |V|$. Now consider an auxilliary *multi-graph* $\Gamma = (\Omega, F)$. Two distinct colourings $\sigma_1, \sigma_2 : V \to [k]$ are adjacent if they only differ at one vertex. In addition we add self-loops to make every vertex have the same degree. A random walk on Γ is equivalent to the following Markov chain $X_0, X_1, \ldots,$ on Ω: Given X_t we generate X_{t+1} as follows:

1. Choose z uniformly at random from V, and c uniformly at random from $\{1, \ldots, k\}$.
2. For all $w \neq z$, set $X_{t+1}(z) = X_t(z)$.
3. If no neighbors of z have color c (i.e., $c \notin X_t(N(z))$), then set $X_{t+1}(z) = c$, otherwise set $X_{t+1}(z) = X_t(z)$.

The version above is called *Metropolis Dynamics*.

Jerrum [29] showed that if $k \geq 2\Delta$ then this chain/walk is rapidly mixing i.e. it gets close to the uniform distribution in time $poly(n)$. From here it is straightforward to devise an algorithm that gives a good approximation to $|\Omega|$ in polynomial time, see [29].

Counting colourings is only a single example of a flourishing research area involving the use of Markov chains to sample from complex distributions and to estimate the size of large combinatorially defined sets. This area of research constitutes an important meeting place for researchers in Statistical Physics and Theoretical Computer Science. For further reading, see [26] or [30].

2.1.2. Expanders In this section we will for convenience asssume that G is a d-regular graph i.e. every vertex has the same degree d. The adjacency matrix $\mathbf{A}$, where $\mathbf{A}(v, w) = 1$ iff v, w are adjacent. $\mathbf{A}$ has largest eigenvalue d and suppose now that λ is its second largest eigenvalue. In this case it is known that

$$T_G(\epsilon) \leq \lceil \frac{\ln \epsilon n}{\ln \lambda} \rceil. \tag{1}$$

We will say that a graph G is an α-expander if for all $S \subseteq V$ with $|S| \leq n/2$, we have $e(S : \bar{S}) \geq \alpha |S|$ where $e(S : \bar{S})$ is the number of edges from S to $\bar{S} = V \setminus S$. In which case one can show that $\lambda \leq 1 - \frac{\alpha^2}{2d^2}$. So if $d = O(1)$ and $\alpha = \Omega(1)$ then a random walk on G mixes in $O(\ln n)$ time. (See Alon [5] and Jerrum and Sinclair [31]. This property of expanders can be used to reduce the number of random bits needed by a randomized algorithm.

We explain the use of the following theorem of Ajtai, Komlós and Szemerédi [1]:

Theorem 1. *Let $G = (V, E)$ be a d-regular graph on n. Let C be a set of cn vertices of G. Then for every ℓ, the number of walks of length ℓ in G that avoid C does not exceed $(1 - c)n((1 - c)d + c\lambda)^\ell$.*

This means that a random walk of length ℓ, with a randomly chosen start vertex, has probability at most $(1 - c)(1 - c(1 - \lambda/d))^\ell < e^{-c(1-\lambda/d)\ell}$ of avoiding C completely. In this context, consider the Miller-Rabin algorithm for testing whether an integer n is prime. Without going into details, it is known that if n is composite then at least $1/2$ of the integers between 1 and n can be used to verify this. So if n is composite and we choose ℓ random integers between 1 and n then the probability we fail to show it is composite is at most $2^{-\ell}$. If we choose our random integers in the normal way, then this requires $\ell \log_2 n$ random bits. On the other hand, suppose that we have a d-regular graph on $[n]$ with $\lambda \leq \epsilon d$ say, and we do a random walk of length ℓ from a randomly chosen start, then this requires $\log_2 n + \ell \log_2 d$ random bits. A significant saving if $d = O(1)$. Applying Theorem 1 we see that the probability we fail to show it is composite is at most $\left(\frac{1+\epsilon}{2}\right)^\ell$.

There are many uses of expanders. See Hoory, Linial and Wigderson [28] for a survey.

2.1.3. Edge Disjoint Paths In this section we discuss the use of random walks to find edge disjoint paths in an expander graph G. We are given a graph $G = (V, E)$ with n vertices, and a set of κ pairs of vertices in V, we are interested in finding for each pair (a_i, b_i), a path connecting a_i to b_i, such that

the set of κ paths so found is edge-disjoint. In general this is an NP-hard problem, but some strong results have been proven in the context of d-regular expanders. In a series of papers, culminating in Frieze [27] it was shown that if G is a sufficiently strong expander and if $\epsilon > 0$ is a sufficently small constant then this problem can be solved if $\kappa \le \epsilon n/\ln n$. First the edge set of G is partitioned to create several edge disjoint expander graphs $G_1, G_2, \ldots G_{10}$. Various phases of the algorithm take place on various G_i. a network flow algorithm is used to connect the a_i to randomly chosen κ-set of vertices by edge disjoint paths. The other endpoint of the path with one endpoint a_i is labelled a_i'. In a similar manner b_i is connected to a randomly chosen b_i'. Finally, a path $a_i', \ldots, x_i, \ldots, b_i'$ is constructed. Here x_i is chosen according to the steady state of a random walk on one of the G_i's and then both a_i' and b_i' are connected to x_i via a random walk.

2.1.4. Randomized Dual Simplex Algorithm In this section we discuss the use of random walks to solve the linear program

LP(b) : minimise cy subject to $\mathbf{A}y = b,\ y \ge 0$. where A is a totally unimodular matrix i.e. all sub-determinants of $\mathbf{A}$ are $0, \pm 1$.

The algorithm described in Dyer and Frieze [23] is somewhat complicated, but it can be viewed as a dual simplex algorithm for the above problem, in which the choice of next pivot is found via a random walk through a geomtrically defined graph. The reader is refered to the paper for details.

2.2. Cover-time:

2.2.1. Log-space algorithm for $s-t$ connecitivity One of the earliest computer science applications of random walk is in [4]. The problem under discussion was whether it is possible to check whether two vertices s and t are in the same component of a bounded degree graph G. The time constraint is polynomial, but the space constraint is logarithmic i.e. only $O(\ln n)$ working storage is allowed. This rules out algorithms like breadth-first and depth-first search. The solution was to consider a random walk from s and run it for $O(mn)$ time. This only requires order $\ln n$ storage and one of the main results of the paper was that the cover time C_G of a connected graph satisfies $C_G \le 2m(n-1)$. Suppose then that we repeat the following ℓ times and still do not reach t. Take a random walk of length $4mn$ from s. If s, t are in the same component then this walk will go through t with probability at least $1 - \frac{2m(n-1)}{4mn} \ge \frac{1}{2}$. So if s, t are indeed in the same component then this algorithm succeeds with probability $\ge 1 - 2^{-\ell}$. It is only recently that Reingold [34] has found a deterministic algorithm that uses LOGSPACE.

2.2.2. Universal Traversal Sequences Suppose that G is a d-regular graph G and that for each vertex $v \in V$ we order the neighbours of v as $x(v,i)$, $i = 1, 2, \ldots, d$. We call this an ordered d-regular graph. Given a start vertex v and a sequence $\sigma = (i_1, i_2, \ldots, i_\ell) \in [d]^\ell$, $\ell = |\sigma|$ we can define a walk $P(v,\sigma) = (v = y_0, y_1, \ldots, y_\ell)$ by $y_{j+1} = x(y_j, i_j)$ i.e. y_{j+1} is the i_jth neighbour of y_j. $P(v,\sigma)$ traverses G if it visits each vertex of G. If we choose σ at random from $[d]^\ell$ and v arbitrarily then the walk $P(v,\sigma)$ is a random walk. Let Z_v denote the time taken by the random walk $\mathcal{W}_v$ to visit all vertices of G. We see then that if $|\sigma| = 4m(n-1)$ then

$$\mathbf{Pr}(\sigma \textit{ does not traverse } G) = \mathbf{Pr}(Z_v \ge 4m(n-1)) \le \mathbf{Pr}(Z_v \ge 2\mathbf{E}(Z_v)) \le \tfrac{1}{2}.$$

Similarly, if $|\sigma| = 4km(n-1)$ then

$$\mathbf{Pr}(\sigma \textit{ does not traverse } G) \le \tfrac{1}{2^k}.$$

Now there are at most n^{dn+1} ways of choosing an ordered d-regular graph and a start vertex. So, with $|\sigma| = 4km(n-1)$,

$$\mathbf{Pr}(\exists \text{ordered } d\text{-regular graph } G \text{ and start vertex } v \text{ such that } \sigma \text{ does not traverse } G) \le \frac{n^{dn+1}}{2^k}. \quad (2)$$

If $k > (dn+1)\ln_2 n$ then the RHS of (2) is less than one. Thus there exists a sequence σ of length $O(dmn \ln n)$ such that for every ordered G and every start vertex v, $P(v,\sigma)$ traverses G. Put another way, using this *universal traversal sequence* we can be sure of ariving at any other vertex, if we follow σ. This being regardless of start vertex v and graph G. So, short (polynomial length) universal traversal sequences exist, but they are very hard to construct explicitly.

2.2.3. Random Spanning Trees Aldous [2] and Broder [10] independently proved the following beautiful result concerning spanning trees of a fixed graph G. Initialise $T = \emptyset$ and start a random walk $\mathcal{W}$ at an arbitrary vertex and when the walk first visits a vertex w add the edge (v,w) to T. Here (v,w) is the edge just traversed by $\mathcal{W}$. The algorithm stops after $\mathcal{W}$ has visited all vertices. The

algorithm generates a spanning tree T of G. The aforementioned papers prove that T is equally likely to be any spanning tree of G. A rather remarkable result.

3. Random Graphs

Various topics arise in the context of random walks on random graphs. Among them are the following: Mixing time of the random walk, cover time of a random graph, properties of multiple particle walks, random walks on graph processes, constructing random networks using random walks.

3.1. Mixing time There is not much to say here except that random graphs tend to be excellent expanders. In some sense they provide the simplest method of generating an expander graph. For example a random walk on an r-regular graph, $r \geq 3$, has mixing time $O(\log n)$ **whp**. In contrast it has proven very difficult to produce *explicit* expanders. It has been done, but the methods can be deep and complicated.

3.2. Cover time of random graphs In this section we study the cover time of various classes of random graphs with fixed vertex set $V = [n] = \{1, 2, ..., n\}$. The spaces of labeled random graphs we consider here are: Erdos-Renyi graphs $G_{n,p}$, random digraphs $D_{n,p}$, random r-regular graphs $\mathcal{G}_r$, preferential attachment graphs $G_m(n)$ and random geometric graphs $G = G(d, r, n)$. A fuller definition of these graph spaces is given below.

It is probably a good time to mention a graph of particular interest in the context of this meeting i.e. *Carbon Nano-Tube* networks, see for example Bush and Li [12]. These graphs are formed from the intersection points of "randomly" placed line segments and one is interested in their *effective resistance*. This parameter is related to commute times, which are related to cover time. We do not have any results yet on a model of such graphs, but it forms a promising line of research.

A few words on notation. Results on random graphs are always asymptotic in n, the size of the vertex set. The notation $A_n \sim B_n$ means that $\lim_{n\to\infty} A_n/B_n = 1$, and **whp** (with high probability) means with probability tending to 1 as $n \to \infty$.

Erdos-Renyi graphs $G_{n,p}$ are defined as follows. The edge $\{i, j\}$ between any pair of vertices i and j occurs with probability p, independently of all other edges. Let C_G denote the vertex cover time. It was shown by Jonasson [32] that **whp**

- $C_G = (1 + o(1))n \ln n$ if $\frac{np}{\ln n} \to \infty$.
- If $c > 1$ is constant and $np = c \ln n$ then $C_G > (1 + \alpha)n \ln n$ for some constant $\alpha = \alpha(c)$.

Thus Jonasson has shown that when the expected average degree $(n - 1)p$ grows faster than $\ln n$, a random graph has the same cover time **whp** as the complete graph K_n, whose cover time is determined by the Coupon Collector problem. Whereas, when $np = \Omega(\ln n)$ this is not the case. This result was refined for sparse graphs as follows:

- If $p = d \ln n/n$ and $d > 1$ then **whp** $C_{G_{n,p}} \sim d \ln\left(\frac{d}{d-1}\right) n \ln n$, [16].
- Let $d > 1$ and let x denote the solution in $(0, 1)$ of $x = 1 - e^{-dx}$. Let X_g be the giant component of $G_{n,p}$, $p = d/n$. Then **whp** $C_{X_g} \sim \frac{dx(2-x)}{4(dx-\ln d)} n(\ln n)^2$, [17]

Considering random r-regular graphs (i.e. the set of all simple r-regular graphs with the uniform measure), we have the following result [14]:
If $G_{n,r}$ denotes a random r-regular graph on vertex set $[n]$ with $r \geq 3$ then **whp** $C_{G_{n,r}} \sim \frac{r-1}{r-2} n \ln n$.
The proof of this result uses a lemma, which we call the first visit time lemma, which under not very restrictive conditions (see e.g. [17]) states that the probability $f(v; T, ..., t)$ that vertex v is not visited by the walk during steps $T, ..., t$ is given by

$$f(v; T, ..., t) = (1 + o(1))(1 - p_v)^t,$$

where T is a mixing time of the walk. Here $p_v \sim \frac{\pi_v}{R_v}$, where π_v is the stationary distribution of vertex v, and R_v is the expected number of returns to v during the mixing time T, of a random walk starting at v. Thus R_v is dependent only on the local geometry of the graph around v. This result is true also for weighted random walks, and general ergodic Markov processes.

The preferential attachment graph $G_m(n)$ is a random graph formed by adding a new vertex at each time step, with m edges which point to vertices selected at random with probability proportional to their degree. Thus at time n there are n vertices and mn edges. This process yields a graph which has been proposed as a simple model of the world wide web [8]. In [15] it is shown that if $m \geq 2$ then **whp** $C_{G_m(n)} \sim \frac{2m}{m-1} n \ln n$.

The random digraphs $D_{n,p}$ are generated in the same manner as $G_{n,p}$ except that now, each directed edge (i, j) occurs independently with probability p. The first visit time lemma applies to these graphs provided they are strongly connected (etc) and we find that: If $p = d \ln n/n$ and $d > 1$ then **whp** $C_{D_{n,p}} \sim d \ln \left(\frac{d}{d-1} \right) n \ln n$.

The main problem for walks on directed graphs is to determine the stationary distribution π_v.

Finally we consider geometric random graphs. Let I denote the unit interval $[0, 1]$ and let $I(d) = [0, 1]^d$ denote the unit torus in d dimensions. We define a random geometric graph $G = G(d, r, n)$ as follows: Sample n points V independently and uniformly at random from $I(d)$ wrapped toroidally. For each point x draw a ball (disk) $D(x, r)$ of radius r about x. The vertex set $V(G) = V$ and the edge set $E(G) = \{\{v, w\} : w \neq v, w \in D(v, r)\}$. The graph serves as a model for ad-hoc networks, where transmitters have limited range.

Avin and Ercal [6] considered the case $d = 2$. They proved that if $G = G(2, r, n)$ and $r^2 > (8 \ln n/n)$ then **whp**$C_G = \Theta(n \ln n)$. For $d \geq 3$ dimensions we can give precise results. Let $G(d, r, n)$, $d \geq 3$ be a random geometric graph. Let $r = (c \ln n)/(\Upsilon_d n)^{1/d}$ and where $c > 1$ is a constant. Then **whp**

$$C_G \sim c \ln \left(\frac{c}{c-1} \right) n \ln n. \tag{3}$$

Here $\Upsilon_d = (\pi^{d/2})/\Gamma(d/2 + 1)$ is the volume of the unit ball $D(0, 1)$ in d dimensions.

3.3. Multiple particle walks Suppose there are $k \geq 1$ particles, each making a simple random walk on a graph G. Essentially there are two possibilities. Either the particles are *Oblivious* or *Interactive*. Oblivious particles act independently of each other, and do not interact on meeting. They may however interact with vertices, possibly in a way determined by previous visits of other particles. Interactive particles, can interact directly in some way on meeting. We assume that interaction only occurs when meeting at a vertex, and that the random walks made by the particles are otherwise independent. Various models and questions arise, e.g.

- **Multiple walks**. For k particles walking independently, we establish the cover time $C_G(k)$ of G.
- **Talkative particles**. For k particles walking independently, which communicate on meeting at a vertex, we study the expected time to broadcast a message.
- **Predator-Prey**. For k predator and ℓ prey particles walking independently, we study the expected time to extinction of the prey particles, when predators eat prey particles on meeting at a vertex.
- **Coalescing particles**. For k particles walking independently, which coalesce on meeting at a vertex, we study the expected time to coalesce to a single particle.
- **Annihilating particles**. For $k = 2\ell$ particles walking independently, which destroy each other (pairwise) on meeting at a vertex, we study the expected time to extinction.

The motivation for these models comes from many sources. Using random walks to test graph connectivity is an established algorithm, and it is appealing to speed up this by parallel searching [11], [7]. Similarly, properties of communication, such as broadcasting and gossiping, between particles moving in a network, is a natural question. In this context, the predator-prey model represents interaction between server and client particles, where each client needs to attach to a server.

Coalescing and annihilating particle systems are part of the classical theory of interacting particles (see e.g. [3]). A system of coalescing particles where initially one particle is located at each vertex, corresponds to another classical problem, the voter model, which is defined as follows: Initially each vertex has a distinct opinion, and at each step each vertex changes its opinion to that of a random neighbour. It is known that the expected time for a unique opinion to emerge, is the same as the

expected time for all the particles to coalesce. By establishing the expected coalescence time, we obtain the expected time for voting to be completed.

The cover time of a random walk on a random r-regular graph was studied in [14], where it was shown with high probability (**whp**), that for $r \geq 3$ the cover time is asymptotic to $\theta_r n \ln n$, where $\theta_r = (r-1)/(r-2)$.

In [20] we prove the following (**whp**) results, arising from the study of multiple random walks on a random regular graph G. For k independent walks on G, the cover time $C_G(k)$ is asymptotic to C_G/k, where C_G is the cover time of a single walk. For most starting positions, the expected number of steps before any of the walks meet is $\theta_r n/\binom{k}{2}$. If the walks can communicate when meeting at a vertex, we show that, for most starting positions, the expected time for k walks to broadcast a single piece of information to each other is asymptotic to $\frac{2\ln k}{k}\theta_r n$, as $k, n \to \infty$.

We also establish properties of walks where there are two types of particles, predator and prey, or where particles interact when they meet at a vertex by coalescing, or by annihilating each other. For example, the expected coalescence time of k particles tends to $2\theta_r n$ as $k \to \infty$; the expected extinction time of k explosive particles (k even) tends to $(2\ln 2)\theta_r n$ as $k \to \infty$. Suppose k predator and ℓ prey particles make random walks, starting in general position (not too near each other). Let $D_{k,\ell}$ be the extinction time of the prey. Then $\mathbf{E}(D_{k,\ell}) \sim \frac{\theta_r H_\ell}{k} n$.

The case of n coalescing particles, where one particle is initially located at each vertex, corresponds to a voter model defined as follows: Initially each vertex has a distinct opinion, and at each step each vertex changes its opinion to that of a random neighbour. The expected time for a unique opinion to emerge is the expected time for all the particles to coalesce, which is asymptotic to $2\theta_r n$.

Combining results from the predator-prey and multiple random walk models allows us to compare expected detection time in the following scenarios: both the predator and the prey move randomly, the prey moves randomly and the predators stay fixed, the predators move randomly and the prey stays fixed. In all cases, with k predators and ℓ prey the expected detection time is $\theta_r H_\ell n/k$, where H_ℓ is the ℓ-th harmonic number. A application of this is with the predators as government agents and the prey as criminals.

3.4. Random walks on random graph processes

If we consider a random graph process $(G(t),\ t = 0, 1, ...)$ in which the graph evolves at each step by the addition of vertices and/or edges then the random walk is searching a growing graph, so we cannot hope to visit all vertices of the graph.

For example, consider a simple model of search, on e.g. the WWW, in which a particle (which we call a spider) makes a random walk on the nodes of an undirected graph process. It is presumed that the spider examines the data content of the nodes for some specific topic. As the spider is walking the graph is growing, and the spider makes a random transition to whatever neighbours are available at the time. For simplicity, we assume that the growth rate of the process and the transition rate of the random walk are similar, so that the spider has at least a chance of crawling a constant proportion of the process. Although the edges of the WWW graph are directed, the idea of evaluating models of search on an undirected process has many attractions, not least its simplicity.

We study the success of the spider's search on comparable graph processes of two distinct types: a random graph process and a web graph process [13]. In the simple process we consider, each new vertex directs m edges towards existing vertices, either choosing vertices randomly (giving a random graph process) or copying according to vertex degree (giving a web graph process). Once a vertex has been added the direction of the edges is ignored.

We consider the following models for the graph process $G(t)$. Let $m \geq 1$ be a fixed integer. Let $[t] = \{1, ..., t\}$ and let $G(1) \subset G(2) \subset \cdots \subset G(t)$. Initially $G(1)$ consists of a single vertex 1 plus m loops. For $t \geq 2$, $G(t+1)$ is obtained from $G(t)$ by adding the vertex t and m randomly chosen edges $\{t+1, v_i\}, i = 1, 2, \ldots, m$, where

Model 1: Vertices $v_1, v_2, \ldots, v_m$ are chosen independently and uniformly with replacement from $[t]$.

Model 2 Vertices $v_1, v_2, \ldots, v_m$ are chosen proportional to their degree after step t. Thus if $d(v, \tau)$ denotes the degree of vertex v in $G(\tau)$ then for $v \in [t]$ and $i = 1, 2, \ldots, m$,

$$\mathbf{Pr}(v_i = v) = \frac{d(v,t)}{2mt}.$$

While vertex t is being added, the spider $\mathcal{S}$ is sitting at some vertex X_{t-1} of $G(t-1)$. After the addition of vertex t, and before the beginning of step $t+1$, the spider now makes a random walk of length ℓ, where ℓ is a fixed positive integer independent of t.

Let $\eta_{\ell,m}(t)$ be the expected proportion of vertices which have not been visited by the spider at step t, when t is large. If we allow $m \to \infty$ we can get precise asymptotic values. Let $\eta_\ell = \lim_{m\to\infty} \eta_{\ell,m}$, then

(a) For Model 1,

$$\eta_\ell = \sqrt{\frac{2}{\ell}} e^{(\ell+2)^2/(4\ell)} \int_{(\ell+2)/\sqrt{2\ell}}^{\infty} e^{-y^2/2}\, dy, \qquad \eta_1 = 0.57\cdots, \text{ and } \eta_\ell \sim 2/\ell \text{ as } \ell \to \infty.$$

(b) For Model 2

$$\eta_\ell = e^\ell 2\ell^2 \int_\ell^\infty y^{-3} e^{-y}\, dy, \qquad \eta_1 = 0.59\cdots, \text{ and } \eta_\ell \sim 2/\ell \text{ as } \ell \to \infty.$$

So for large m, t and $\ell = 1$ it is slightly harder for the spider to crawl on a webgraph whose edges are generated by a copying process (Model 2) than on a uniform choice random graph (Model 1).

3.5. Constructing random networks using random walks

Bourassa and Holt [9] propose a decentralised protocol for P2P networks based on random walks. If a vertex in the network needs an address of a random vertex, then it initiates a random walk and gets the address of the vertex reached at some specified step of the walk. The protocol constructs a 4-regular random graph. Their protocol, however, cannot reconnect the network if it becomes disconnected.

In [21] we describe a randomized algorithm for assigning neighbours to vertices joining a P2P network. The aim of the algorithm is to maintain connectivity, low diameter and constant vertex degree. On joining each vertex donates a constant number c of tokens to the network. These tokens contain the address of the donor vertex. Tokens make independent random walks in the network. A token can be used by any vertex it is visiting, to establish a connection to the donor vertex. This allows vertices which initially join in an arbitrary manner (e.g to a friend/super-node) to be re-allocated to a random set of neighbours although the overall vertex membership of the network is unknown. The new vertex joins arbitrarily, collects m tokens, attaches to the vertices whose addresses they contain and detaches from its original contacts.

If t is the size of the network, then the diameter of the network is $O(\ln t)$ for all t, with high probability. The network is extremely robust under adversarial deletion of vertices and edges and actively reconnects itself when broken. As an example of the robustness of this model, suppose an adversary deletes edges from the network leaving components of size at least $t^{1/2+\delta}, \delta > 0$ small. With high probability the network rapidly reconnects itself by replacing lost edges using tokens from the token pool.

References

1. M. Ajtai, J. Komlós and Szemerédi, Deterministic simulation in LOGSPACE, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing* (1987) 132-140.
2. D. Aldous, The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees, *SIAM Journal on Discrete Mathematics* 3 (1990) 450-465.
3. D. Aldous and J. Fill. Reversible Markov Chains and Random Walks on Graphs, http://stat-www.berkeley.edu/pub/users/aldous/RWG/book.html.
4. R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász and C. Rackoff, Random Walks, Universal Traversal Sequences, and the Complexity of Maze Problems. *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science* (1979) 218-223.
5. N. Alon, Eigenvalues and expanders, *Combinatorica* 6 (1986) 83-96.

6. C. Avin and G. Ercal, On the cover time and mixing time of random geometric graphs, *Theoretical Computer Science* 380 (2007) 2-22
7. N. Alon, C. Avin, M. Kouchý, G. Kozma, Z. Lotker, M. Tuttle. Many random walks are faster then one, to appear.
8. A. Barabási and R. Albert, Emergence of scaling in random networks, *Science* 286 (1999) 509-512.
9. V. Bourassa and F. Holt, SWAN: Small-world wide area networks, in Proceedings of International Conference on Advances in Infrastructure (SSGRR 2003w), L'Aquila, Italy, 2003, paper # 64.
Available at `http://www.panthesis.com/content/swan_white_paper.pdf`
10. A.Z. Broder, Generating Random Spanning Trees, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science* (1989) 442-447
11. A. Broder, A. Karlin, A. Raghavan and E. Upfal. Trading space for time in undirected s-t connectivity. Proc. ACM Symp. Theory of Computing (1989) 543-549.
12. S.F. Bush and Y. Li, Network Characteristics of Carbon nanotubes: A Graph Eigenspectrum Approach and Tool Using Mathematica, GE Technical Information Series Report: 2006GRC023.
13. C. Cooper, A. Frieze. Crawling on simple models of web-graphs. *Internet Mathematics* 1, 57-90 (2003).
14. C. Cooper and A. M. Frieze, The cover time of random regular graphs, *SIAM Journal on Discrete Mathematics*, 18 (2005) 728-740.
15. C. Cooper and A. M. Frieze, The cover time of the preferential attachment graph. *Journal of Combinatorial Theory Series B*, Series B, 97(2) 269-290 (2007).
16. C. Cooper and A. M. Frieze, The cover time of sparse random graphs, *Random Structures and Algorithms* 30 (2007) 1-16.
17. C. Cooper and A. M. Frieze, The cover time of the giant component of a random graph, *Random Structures and Algoritms*, 32, 401-439 (2008).
18. C. Cooper and A. M. Frieze, The cover time of random digraphs, Proceedings of RANDOM 2007, (11th. International Workshop on Randomized Techniques in Computation).
19. C. Cooper and A. M. Frieze, The cover time of random geometric graphs, to appear.
20. C. Cooper, A. Frieze, T. Radzik. Multiple random walks in random regular graphs, to appear.
21. C. Cooper, R. Klasing, T. Radzik. A randomized algorithm for the joining protocol in dynamic distributed networks. *J. Theoretical Computer Science* (to appear)
Also published as INRIA report RR-5376 and CNRS report I3S/RR-2004-39-FR
22. P.G. Doyle and J. Snell, Random Walks and Electric Networks, Mathematical Association of America, 1984. (Now available from http://xxx.lanl.gov/abs/math.PR/0001057.
23. M.E. Dyer and A. M. Frieze, Random walks, totally unimodular matrices and a randomised dual simplex algorithm, *Mathematical Programming* 64 (1994) 1-16.
24. U. Feige, A tight upper bound for the cover time of random walks on graphs, *Random Structures and Algorithms*, 6 (1995) 51-54.
25. U. Feige, A tight lower bound for the cover time of random walks on graphs, *Random Structures and Algorithms*, 6 (1995) 433-438.
26. http://www.math.cmu.edu/ af1p/Mixingbook.pdf
27. A.M. Frieze, Edge disjoint paths in expander graphs, *SIAM Journal on Computing* 30 (2001) 1790-1801.
28. S. Hoory, N. Linial and A. Wigderson, Expander graphs and their applications, *Bulletin of the American Mathematical Society* 43 (2006) 439-561.
29. M.R. Jerrum, A very simple algorithm for estimating the number of k-colourings of a low-degree graph, *Random Structures and Algorithms* 7(2):157–165, 1995.
30. M.R. Jerrum, *Counting, sampling and integrating: algorithms and complexity*, Lectures in Mathematics–ETH Zürich, Birkhäuser, 2003.
31. M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems.* (D. Hochbaum ed.) PWS (1996) 482-520
32. J. Jonasson, On the cover time of random walks on random graphs, *Combinatorics, Probability and Computing*, 7 (1998), 265-279.
33. G. Polya, Über eine Aufgabe der Wahrscheinlichkeitstheorie betreffend die Irrfahrt im Strassennetz, Mathematische Annalen, 84 (1921) 149-160.
34. O. Reingold, Undirected ST-Connectivity in Log-Space, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing* (2005) 376-385.