A Fast Algorithm to Find All High-Degree Vertices in Graphs with a Power-Law Degree Sequence

Colin Cooper, Tomasz Radzik, and Yiannis Siantos

Abstract. We develop a fast method for finding all high-degree vertices of a connected graph with a power-law degree sequence. The method uses a biased random walk, where the bias is a function of the power law c of the degree sequence.

Let $G(t)$ be a t-vertex graph, with degree sequence power law $c \geq 3$ generated by a generalized preferential attachment process that adds m edges at each step. Let S_a be the set of all vertices of degree at least t^a in $G(t)$. We analyze a biased random walk that makes transitions along undirected edges $\{x, y\}$ with probabilities proportional to $(d(x)d(y))^b$, where $d(x)$ is the degree of vertex x and $b > 0$ is a constant parameter. With parameter $b = (c-1)(c-2)/(2c-3)$, the random walk discovers the set S_a completely in $\widetilde{O}(t^{1-2ab(1-\epsilon)})$ steps with high probability. The error parameter ϵ depends on c, a, and m.

The cover time of the entire graph $G(t)$ by the biased walk is $\widetilde{O}(t)$. Thus the expected time to discover all vertices by the biased walk is not much higher than the $\Theta(t \log t)$ cover time of a simple random walk.

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

ISSN: 1542-7951 print

The standard preferential attachment process generates graphs with power law $c = 3$. The search parameter $b = 2/3$ is appropriate for such graphs. We conduct experimental tests on a preferential attachment graph and on a sample of the underlying graph of the World Wide Web with power law $c \approx 3$ that support the claimed property.

1. Introduction

Many large networks have a heavy-tailed degree sequence. Thus, although the majority of the vertices have constant degree, a very distinct minority have very large degrees. This particular property is the significant defining feature of such graphs. A log-log plot of the degree sequence breaks naturally into three parts: the lower range (small constant degree), where there may be curvature because the power law approximation is incorrect; the middle range, of large but well-represented vertex degrees that give the characteristic straight-line log-log plot of the power-law coefficient; and the upper tail, where the sequence is far from concentrated and the plot is a spiky mess. See, for example, Figure 1 (the degree sequence of a simulated preferential attachment graph with $m = 4$ edges added at each step) and Figure 3 (the degree sequence of the underlying graph of a sample of the World Wide Web). In both cases, the horizontal axis is $a = \log d/ \log t$, where d is vertex degree, and t is the size of the graph.

In this paper, we focus on sampling higher-degree vertices, including those in the middle range as well as the upper tail. Our aim is to find all these vertices, and we propose a provably efficient method of obtaining them in sublinear time using a weighted random walk. One reason for finding all the higher-degree vertices is that the upper tail is not concentrated, so no subsample will be representative. We consider a weighted random walk because since there are few vertices even in the middle range, a simple random walk may take too long to obtain a statistically significant sample. Coupled with this is the impression that in many networks, for example the World Wide Web, it is the high-degree vertices that are important, both as hubs and authorities, and for PageRank calculations.

Our eventual aim is to devise search algorithms based on weighted random walks that quickly locate all high-degree vertices in arbitrary connected graphs with a heavy-tailed degree sequence. To do this, we use a random walk with transition probability along edge $\{x, y\}$ proportional to $(d(x)d(y))^b$, where $d(x)$ is the degree of vertex x, and b is a positive constant.

The paper is divided into two parts. In the first, we make a detailed investigation of the best kind of weighted random walk to find high-degree vertices in a graph generated using the preferential attachment model, and with known degree-sequence power law c. In this case, we find the optimal value $b = b(c)$ to use in the transition probabilities of the weighted random walk. The theory is developed in Sections 2–5, and the experimental results given in Section 6. Because we use weighted random walks whose mixing properties are not known, the analysis we make for the preferential attachment model does not use notions of mixing time or stationary distribution.

In the second part of the paper, in Section 7, we discuss what can be obtained in the case of general heavy-tailed degree sequences when the power law c is unknown. Our analysis in Section 7 uses notions of mixing time and stationary distribution, and we quantify things as best we can. In particular, we develop reasoning to suggest that a weight of $b = 1/2$ could be used as a search heuristic and give experimental evidence for the effectiveness of this choice.

Previous work on efficient sampling of network characteristics has arisen in many areas. In the context of search engine design, studies in optimally sampling the URL crawl frontier to rapidly sample, for example, high-PageRank vertices, based on knowledge of vertex degree in the current sample, can be found, for instance, in [Baeza-Yates et al. 05].

Within the random graph community, *traceroute sampling* was used to estimate cumulative degree distributions, and methods of removing the high-degree bias from this process were studied, for example, in [Achlioptas et al. 09, Flaxman and Vera 07]. Another approach, analyzed in [Brautbar and Kearns 10], is the *jump and crawl* method to find, for instance, all vertices of very high degree. The method uses uniform sampling followed by inspection of the neighboring vertices, in a time sublinear in the network size.

In the context of online social networks, exploration has often focused on how to discover the entire network more efficiently. Until recently, this was feasible for many real-world networks before they exploded to their current sizes. It is no longer feasible to get a consistent snapshot of the Facebook network, for example. According to the Facebook statistics page[1] retrieved on June 2, 2011, there were over 500 million active Facebook users and around 36 billion links.

Methods based on random walks are commonly used for graph searching and crawling. In [Stutzbach et al. 06], the performance of breadth-first search (BFS) with a simple random walk is compared with a Metropolis–Hastings random walk on various classes of random graphs as a basis for sampling the degree distribution of the underlying networks.

The purpose of the investigation was to sample from dynamic peer-to-peer (P2P) networks. In a related study [Gjoka et al. 09], the authors made

[1] www.facebook.com/press/info.php?statistics.

extensive use of these methods to collect a sample of Facebook users. Since simple random walks are degree biased, they used a reweighting technique to unbias the sampled degree sequence output by the random walk. This is referred to in [Gjoka et al. 09] as a reweighted random walk. In both the above cases, it was shown that bias could be removed dynamically using a suitable Metropolis–Hastings random walk.

2. Preferential Attachment Graphs, Models, and Search Methods

A simple way to generate a graph with a power-law degree sequence is to use the preferential attachment method described in [Barabási and Albert 99]. In this model, the graph $G(t) = G(m,t)$ is obtained from $G(t-1)$ by adding a new vertex v_t with m edges between v_t and $G(t-1)$. The endpoints of these edges are chosen preferentially, that is, to be proportional to the existing degree of vertices in $G(t-1)$. Thus the probability $p(x,t)$ that vertex $x \in G(t-1)$ is chosen as the endpoint of a given edge is equal to $p(x,t) = d(x,t-1)/(2m(t-1))$, and this choice is made independently for each of the m edges added. A model generated in this way has a power law of $c = 3$ for the degree sequence, irrespective of the number of edges $m \geq 1$ added at each step. For a graph constructed in this way, the expected degree at step t of the vertex added at step s is $\mathbb{E}\ d(s,t) \sim m(t/s)^{1/2}$.

The preferential attachment model was refined in [Bollobás et al. 01], whose authors introduced the scale-free model to make detailed calculations of degree sequences. The model was generalized by many authors, including the web-graph model of [Cooper and Frieze 03]. The web-graph model allows the number of edges added at each step to vary, for edges from new vertices to choose their endpoints preferentially or uniformly at random, and for insertion of edges between existing vertices. By varying these parameters, preferential attachment graphs with degree sequences exhibiting power laws c in the interval $(2,\infty)$ are obtained.

The power law c for preferential attachment graphs and web graphs can be written explicitly as

$$c = 1 + \frac{1}{\eta}, \tag{2.1}$$

where η is the expected proportion of edge endpoints added preferentially (see [Cooper 06]). For example, in the standard preferential attachment process (e.g., the Barabási and Albert model), $\eta = 1/2$, since each new edge chooses an existing neighbor vertex preferentially, thus explaining the power law of 3 for this model.

In the simplest case, to form $G(t)$, a new vertex v_t is added at each step t with m edges directed toward the existing graph $G(t-1)$. Each edge chooses its terminal vertex either by preferential attachment or uniformly at random with some probability mixture p or $1-p$. This generates a power law $c \geq 3$. We refer to this generalized process as $G(c, m, t)$. For this example, the parameter $\eta = p/2$ depends on the proportion p of edge endpoints chosen preferentially (as opposed to uniformly at random).

A straightforward way to generate graphs $G(c, m, t)$ with m edges added at each step and with power-law parameter $2 < c < 3$ is as follows. At each step, either add a new vertex with probability α and attach the m edges preferentially to existing vertices, or with probability $1-\alpha$ insert the m edges between existing vertices, choosing the initial and terminal endpoints of the edges preferentially. This gives a parameter $\eta = 1 - \alpha/2$.

The parameter η in (2.1) occurs in process models in the expression for the expected degree of a vertex. Let $d(s,t)$ denote the degree at step t of the vertex v_s added at step s. The expected value of $d(s,t)$ is given by

$$\mathbb{E}\ d(s,t) \sim m \left(\frac{t}{s}\right)^{\eta}. \tag{2.2}$$

Thus, in the preferential attachment model of [Barabási and Albert 99], $\mathbb{E}\ d(s,t) \sim m(t/s)^{1/2}$.

To generalize this, we consider multigraphs $G(t)$ on t vertices with properties (i) and (ii) below, where $\epsilon > 0$ and $0 < \eta < 1$ are parameters. We call such graphs *pseudopreferential*.

(i) When the vertices are relabeled $s = 1, \ldots, t$ by sorting on vertex degree in descending order, $G(t)$ has a degree sequence that satisfies, for each $s \geq 1$,

$$\left(\frac{t}{s}\right)^{\eta(1-\epsilon)} \leq d(s) \leq \left(\frac{t}{s}\right)^{\eta} \log^2 t. \tag{2.3}$$

(ii) For all vertices s in the sorted order, s has at most m edges to vertices $\sigma \leq s$.

Our particular aim is, given $a > 0$, to find all vertices $v \in V(t)$ of degree $d(v) \geq t^a$. Denote by S_a the set of vertices of $G(t)$ of degree $d(v) \geq t^a$. For the following reason, we will assume $a < \eta$. The maximum degree in (2.3) is $\widetilde{O}(t^{\eta})$, and from (2.2), this is also the maximum expected degree in preferential attachment graphs ($\eta = 1/2$) and web graphs ($0 < \eta < 1$). We use the notation $\widetilde{O}(f(t))$ as shorthand for $O(f(t) \log^k t)$, where t is the size of $G(t)$ and k is a positive constant.

When we apply a random walk to find all vertices in a set S, we say that the walk is *seeded* if it starts from some vertex of S. In the context of searching networks such as Facebook, Twitter, or the World Wide Web, it is not unreasonable to suppose we know some high-degree vertex without supposing we know all of them. Experimentally, we found the seeding condition unnecessary, but a general analysis without this condition would require notions of mixing time and stationarity that our analysis avoids. The following theorem holds for every network with the pseudopreferential properties given above.

Theorem 2.1. *Let $G(t)$ be a pseudopreferential graph with degree sequence satisfying* (2.3)*. Let $S_a = \{v : d(v) \geq t^a\}$ be connected with diameter* $\mathrm{Diam}(S_a)$*. Let*

$$b = \frac{1-\eta}{\eta(2-\eta(1-\epsilon))}.$$

A biased seeded random walk with transition probability along edge $\{x, y\}$ proportional to $(d(x)d(y))^b$ finds all vertices in $G(t)$ of degree at least t^a in $\widetilde{O}(\mathrm{Diam}(S_a) \times t^{1-2ab})$ steps, with high probability (whp). The cover time of the graph $G(t)$ by this biased walk is $\widetilde{O}(t\,\mathrm{Diam}(G(t)))$.

In reality, the degree sequence (2.3) of graph $G(t)$ is unknown, but η can be estimated as $\eta = 1/(c-1)$ from the power law c of the degree sequence if that is known. Optimistically setting $\epsilon = 0$ gives a value b for the search algorithm. It is also fair to say that experimentally, we found that putting $b = 1/2$ in the biased random walk was effective for a variety of real networks with a power-law degree sequence.

We next give a general result for web graphs $G(c, m, t)$ that is also valid for related models such as scale-free graphs. For the class of graphs $G(c, m, t)$, the lower bound on the degree of vertex s becomes less concentrated as s tends to t, so that the value of ϵ we must choose for our lower bound in (2.3) increases with s. Thus, as the vertex degree t^a decreases, the upper bound on the algorithm's running time increases in a way that depends on a, c, m. As long as we incorporate this dependence, Theorem 2.2 says that if we search $G(c, m, t)$ using a random walk with a bias b proportional to the power law c, then we can find all high-degree vertices quickly, and the time to discover all vertices is of about the same order as for a simple random walk.

Theorem 2.2. *Let $c \geq 3$, $m \geq 2$, $a < 1/(c-1)$, and $\epsilon = (1 + 1/a - c)/m$. Let $b = (c-1)(c-2)/(2c-3)$.*

A biased seeded random walk with transition probability along edge $\{x, y\}$ proportional to $(d(x)d(y))^b$ finds all vertices in $G(c, m, t)$ of degree at least t^a in $\widetilde{O}\left(t^{1-2ab(1-\epsilon)}\right)$ steps whp. The cover time of the graph $G(c, m, t)$ by this biased walk is $\widetilde{O}(t)$.

The maximum degree of $G(c, m, t)$ is $\widetilde{O}(t^\eta)$ whp, where $\eta = 1/(c-1)$, which explains the bound on a given above. Using this, a t^{1-2ab} running time can be repackaged as follows: Let $a = \theta\eta$ for $0 < \theta < 1$. Then $2ab = \theta(1 - 1/(2c-3))$.

3. Properties of the Web-Graph Process

The actual value of $d(s, t)$ is not concentrated around $\mathbb{E}\ d(s, t)$ in the lower tail, but the following lemma proved in [Cooper 06] is adequate for our analysis.

Lemma 3.1. *Given $G(c, m, t)$ and a, ϵ, suppose*

$$m > \frac{1}{\epsilon}\left(\frac{1}{a} - \frac{1}{\eta}\right),$$

where η is as in (2.1).

With high probability, for all vertices s such that $\mathbb{E}\ d(s, t) \geq t^a$, we have that

$$d(s, t) \geq \left(\frac{t}{s}\right)^{\eta(1-\epsilon)}.$$

For all $s \geq \log^2 t$, we have $d(s, t) \leq \left(\frac{t}{s}\right)^\eta \log^2 t$. For all $s \geq 1$, we have $d(s, t) = \widetilde{O}(t^\eta)$.

The upshot of this is that all vertices added after step $v = s \log^{2/\eta+1} t$ have degree $d(v, t) = o\left((t/s)^\eta\right)$ whp. This observation forms the basis of our sublinear algorithm.

We also need lower tail concentration for large sets of vertices.

Lemma 3.2. *Let $d([s], t)$ denote the degree of $[s] = \{1, \ldots, s\}$ at step t. Let $K > 1$. Then*

$$\Pr\left(d([s], t) \leq \frac{2ms}{K}\left(\frac{t}{s}\right)^\eta\right) = O\left(s^{-mK}\right).$$

Proof. We give the proof for $\eta = 1/2$ (preferential attachment), the general proof being similar.

Let $Z_t = d([s], t)$. Then $Z_s = 2ms$, since m edges are added in the first s steps. From step $t \geq s + 1$, we have $Z_t = X_t + Z_{t-1}$, where X_t is the binomial random variable:

$$X_t \sim Bin\left(m, \frac{Z_{t-1}}{2m(t-1)}\right).$$

It follows that $\mathbb{E}\ Z_t \sim 2ms(t/s)^{1/2}$.

Given $h, c_t, A > 0$, we have

$$\Pr(Z_t < A) = \Pr\left(e^{-hZ_t/c_t} > e^{-hA/c_t}\right).$$

Let $p = Z_{t-1}/(2m(t-1))$. Then

$$\mathbb{E}\left(e^{-hX_t/c_t}\right) = \left(1 - p + pe^{-h/c_t}\right)^m \le e^{-\frac{h}{c_t}(1-h/c_t)\frac{Z_{t-1}}{2(t-1)}},$$

where we have used that $e^{-x} \le 1 - x + x^2$. Let $c_s = 1$ and

$$c_t = \left(1 + \frac{1}{2(t-1)}\right) c_{t-1},$$

so that $c_t \sim (t/s)^{1/2} \sim \mathbb{E}\ Z_t/(2ms)$. We will choose $h = o(1)$ (see below). Iterating the expression $Z_t = X_t + Z_{t-1}$ and conditioning on the value of Z_{t-1} gives

$$\mathbb{E}\,(e^{-hZ_t/c_t} \mid Z_{t-1}) \le \exp\left(-h\frac{Z_{t-1}}{c_{t-1}} \frac{1 + (1 - h/c_t)/(2(t-1))}{1 + 1/(2(t-1))}\right) = e^{-h'Z_{t-1}/c_{t-1}}, \tag{3.1}$$

where

$$h\left(1 - O\left(\frac{h}{tc_t}\right)\right) \le h' \le h$$

and

$$\mathbb{E}\,(e^{-hZ_s/c_s}) = e^{-h2ms}.$$

The conditioning in (3.1) can be done by backward recursion by taking the expectation of Z_{t-1} conditional on Z_{t-2}, and so on, until we arrive at the expectation of Z_s, which is constant. Thus

$$\mathbb{E}\,(e^{-hZ_t/c_t}) \le \mathbb{E}\left(\exp\left(-h\frac{Z_s}{c_s}\prod_{j=s}^{t-1}(1 - O\left(\frac{h}{jc_j}\right)\right)\right) = e^{-h2ms(1-O(h))}.$$

Applying the Markov inequality $\Pr(Y \ge A) \le \mathbb{E}\,(Y)/A$ with $Y = e^{-hZ_t/c_t}$ and $A = e^{-h\,\mathbb{E}\,(Z_t)/(Kc_t)} \sim e^{-2hms/K}$, we have

$$\begin{aligned}\Pr(Z_t \le \mathbb{E}\ Z_t/K) &= \Pr\left(e^{-hZ_t/c_t} \ge e^{-h\,\mathbb{E}\,(Z_t)/(Kc_t)}\right) \le e^{-h2ms(1-1/K-O(h))}\\ &= O(s^{-mK}),\end{aligned}$$

on choosing $h = (K \log s)/s = o(1)$. □

Another piece of the puzzle we need is that whp, web graphs have diameter

$$\mathrm{Diam}(G(c, m, t)) = O(\log t). \tag{3.2}$$

Detailed investigations of the diameter of preferential attachment graphs have appeared in the literature, including [Bollobás and Riordan 04]. Experimental investigations of average distances in the World Wide Web are given in [Barabási et al. 00, Broder et al. 00]. A crude proof of the $O(\log t)$ upper bound can be made using the expansion properties of the graph. For example, in the preferential attachment graph ($\eta = 1/2$, $c = 3$), when vertex t is added to $G(m, t)$, the probability that t does not select at least one neighbor in $G(t/2)$ is at most

$$\left(1 - \frac{2m(t/2)}{2mt}\right)^m = \left(\frac{1}{2}\right)^m .$$

Thus we see that $\mathrm{Diam}(G(m, t)) = O(\log t)$ by a "tracing backward stochastically" argument.

4. Biased Random Walks

Let $G = (V, E)$ be a connected undirected graph. A *random walk* W_u, $u \in V$, on G is a Markov chain $X_0 = u, X_1, \ldots, X_t, \ldots$ on the vertices V associated with a particle that moves from vertex to vertex according to a transition rule. The probability of a transition from vertex i to vertex j is $p(i, j)$ if $\{i, j\} \in E$, and 0 otherwise.

Let $d(v) = d(v, t)$ be the degree of vertex $v \in G(t)$, and let $N(v)$ denote the neighbors of v in this graph. The basis of our algorithm is a degree-biased random walk, with transition probability $p(u, v)$ given by

$$p(u, v) = \frac{(d(v))^b}{\sum_{w \in N(u)} (d(w))^b}, \tag{4.1}$$

where $b > 0$ is a constant. The value of b that we will choose in our proofs is optimized to depend on η. Thus for Theorem 2.2, the value of b can be expressed directly using (2.1) as a function of the degree-sequence power law c.

The easiest way to reason about biased random walks is to give each edge e a weight $w(e)$, so that transitions along edges are made proportional to this weight. In the case above, the weight of the edge $e = (u, v)$ is given by $w(e) = (d(u)d(v))^b$, so that the transition probability (4.1) is now written as

$$p(u, v) = \frac{(d(u)d(v))^b}{\sum_{w \in N(u)} (d(u)d(w))^b}. \tag{4.2}$$

The inspiration for the degree-biased walk above comes from the β-walks of [Ikeda et al. 03], which use an edge weight $w(x, y) = 1/(d(x)d(y))^\beta$ to favor low-degree vertices. When $\beta = 1/2$, this gives an improved worst-case bound of $O(n^2 \log n)$ for the cover time of connected n-vertex graphs.

We next note some facts about weighted random walks, which can be found in [Aldous and Fill 95] or [Lovász 96]. The weight $w(e)$ of an edge e has the meaning of conductance in electrical networks, and the resistance $r(e)$ of e is given by $r(e) = 1/w(e)$. The commute time $K(u, v)$ between vertices u and v is the expected number of steps taken to travel from u to v and back to u. The commute time for a weighted walk is given by

$$K(u, v) = w(G)R_{\mathsf{eff}}(u, v). \tag{4.3}$$

Here $w(G) = 2\sum_{e\in E} w(e)$ and $R_{\mathsf{eff}}(u, v)$ is the effective resistance between u and v when G is taken as an electrical network with edge e having resistance $r(e)$. For our proof, we do not need to calculate $R_{\mathsf{eff}}(u, v)$ very precisely, but rather, we note that if uPv is any path between u and v, then

$$R_{\mathsf{eff}}(u, v) \le \sum_{e\in uPv} r(e).$$

For $u \in V$ and a subset of vertices $S \subseteq V$, let $C_u(S)$ be the expected time taken for W_u to visit every vertex of S. The *cover time* C_S of S is defined as $C_S = \max_{u\in V} C_u(S)$. We define a walk as *seeded* if it starts in S. The *seeded cover time* C^*_S of S is defined as $C^*_S = \max_{u\in S} C_u(S)$. For a random walk starting in a set S, the cover time of S satisfies the following Matthews bound:

$$C^*_S \le \max_{u,v\in S} H(u, v) \log |S|. \tag{4.4}$$

This is obtained by restricting the proof of the Matthews bound (see, e.g., [Aldous and Fill 95, Chapter 2, Theorem 26]) to the set S and to a random walk starting in the set S. For $u \ne v$, the variable $H(u, v)$ is the expected time to reach v starting from u (the hitting time). The commute time $K(u, v)$ is given by $K(u, v) = H(u, v) + H(v, u)$, so $K(u, v) > H(u, v)$.

5. Proof of Theorems 2.1 and 2.2

We apply the Matthews bound (4.4) to the set S_a. Clearly, $\log |S_a| \le \log t$. It remains to find

$$\max_{u,v\in S_a} H(u, v) \le \max_{u,v\in S_a} K(u, v).$$

To calculate $K(u, v)$ in (4.3), we first need to bound $w(G)$.

Lemma 5.1. *By choosing*

$$b = \frac{1-\eta}{\eta(2-\eta(1-\epsilon'))},$$

where $\epsilon' = \epsilon$ for the pseudopreferential graphs (Theorem 2.1) and $\epsilon' = 0$ for the web graphs (Theorem 2.2), it follows that $w(G) = \widetilde{O}(t)$.

Proof. The proofs for the pseudopreferential graphs and the web graphs are very similar, so we present them together. For the web graphs, all statements should be understood as holding with high probability.

We define a graph G^* on vertices $1, 2, \ldots, t$ that has the same degree sequence as graph G and is built in a similar iterative process: for each $v = t_0, t_0+1, \ldots, t$, add m edges from vertex v to some earlier vertices. In graph G, edges are selected according to a random preferential process, while in graph G^*, it is done according to the deterministic process that greedily fills the in-degrees of vertices, giving preference to the older vertices. In both graphs, if (x, y) is a directed edge, then $y < x$ (the edges point from x toward the earlier vertex y).

Assume $b > 0$ and define

$$\bar{d}(v) = \left(\frac{t}{v}\right)^{\eta}, \quad \bar{w}(G) = 2 \sum_{\{x,y\}\in E(G)} \left(\bar{d}(x)\bar{d}(y)\right)^{b}.$$

Using (2.3) for the pseudopreferential graphs and Lemma 3.1 for the web graphs, we see that in both graphs, the degree of vertex v is

$$d(v) = \bar{d}(v) \cdot O(\mathrm{polylog}(t)),$$

so we have

$$w(G) = \bar{w}(G) \cdot O(\mathrm{polylog}(t)). \tag{5.1}$$

Graph G^* is obtained from G by repeatedly swapping edges. Whenever there is a pair of edges (x, y), (u, v) such that $x < u$ but $y > v$, then replace them with edges (x, v) and (u, y). If $A > B$ and $C > D$, then $(A-B)(C-D) > 0$, so $AC + BD > AD + BC$. Thus each swap increases $\bar{w}(G)$, because

$$(\bar{d}(x))^b > (\bar{d}(u))^b \quad \text{and} \quad (\bar{d}(y))^b < (\bar{d}(v))^b$$

implies

$$(\bar{d}(x))^b(\bar{d}(v))^b + (\bar{d}(u))^b(\bar{d}(y))^b > (\bar{d}(x))^b(\bar{d}(y))^b + (\bar{d}(u))^b(\bar{d}(v))^b.$$

Therefore,

$$\bar{w}(G^*) \geq \bar{w}(G). \tag{5.2}$$

By construction, a vertex v in G^* has incoming edges originating from consecutive vertices $\text{first}(v), \text{first}(v)+1, \ldots, \text{last}(v)$. Thus we have

$$\begin{aligned}\bar{w}(G^*) &= 2 \sum_{\{y,x\} \in E(G^*)} \left(\bar{d}(x)\bar{d}(y)\right)^b = 2 \sum_{x=1}^{t} \sum_{y=\text{first}(x)}^{\text{last}(x)} \left(\bar{d}(x)\bar{d}(y)\right)^b \\ &\leq 2 \sum_{x=1}^{t} d(x) \left(\bar{d}(x)\bar{d}(\text{first}(x))\right)^b \\ &= O(\text{polylog}(t)) \cdot \sum_{x=1}^{t} \left(\bar{d}(x)\right)^{1+b} \left(\bar{d}(\text{first}(x))\right)^b . \end{aligned} \tag{5.3}$$

Now we bound $\text{first}(x)$. There are at most $m \cdot \text{first}(x)$ edges outgoing from vertices $1, 2, \ldots, \text{first}(x)$, and these edges fully fill the in-degrees of vertices $1, 2, \ldots, x-1$, so

$$m \cdot \text{first}(x) \geq \sum_{z=1}^{x-1} d_{in}(z) = \sum_{z=1}^{x-1} (d(z) - d_{\text{out}}(z)) \geq \sum_{z=1}^{x-1} (d(z) - m).$$

Hence

$$\text{first}(x) \geq \frac{1}{2m} \sum_{z=1}^{x-1} d(z).$$

Let C be some generic constant whose value can vary. For Theorem 2.1, choosing $\epsilon' = \epsilon$, we have for all $x \geq 1$,

$$\sum_{z=1}^{x-1} d(z) \geq \sum_{z=1}^{x-1} \left(\frac{t}{z}\right)^{\eta(1-\epsilon')} = C t^{\eta(1-\epsilon')} x^{1-\eta(1-\epsilon')} = C x \left(\frac{t}{x}\right)^{\eta(1-\epsilon')} .$$

For Theorem 2.2 (web-graph case), choosing $\epsilon' = 0$, we have from Lemma 3.2 that for all $x \geq \log t$,

$$\sum_{z=1}^{x-1} d(z) \geq m x \left(\frac{t}{x}\right)^{\eta(1-\epsilon')} .$$

Thus in both cases, for all $x \geq \log t$,

$$\bar{d}(\text{first}(x)) = \left(\frac{t}{\text{first}(x)}\right)^{\eta} \leq C \left(\frac{t}{x}\right)^{\eta(1-\eta(1-\epsilon'))} . \tag{5.4}$$

Using (5.3), (5.4), and the bound $\bar{d}(\text{first}(x)) = \widetilde{O}(t^{\eta})$ for $x \leq \log t$, we get

$$\begin{aligned}\bar{w}(G^{*}) &= O(\text{polylog}(t)) \cdot \sum_{x=1}^{t} \left(\frac{t}{x}\right)^{\eta(1+b)} \left(\frac{t}{x}\right)^{b\eta(1-\eta(1-\epsilon'))} \\ &= O(\text{polylog}(t)) \cdot \sum_{x=1}^{t} \left(\frac{t}{x}\right)^{\eta(1+b(2-\eta(1-\epsilon')))} . \qquad (5.5)\end{aligned}$$

Choosing

$$\eta(1 + b(2 - \eta(1 - \epsilon'))) = 1, \qquad (5.6)$$

we see that the sum in (5.5) is $O(t \log t)$. This, (5.1), and (5.2) imply $w(G) = \widetilde{O}(t)$. □

Recall that we analyze the seeded cover time of the set $S_a = \{v : d(v,t) \geq t^a\}$.

Details specific to Theorem 2.1. The set S_a is connected with diameter $\text{Diam}(S_a)$, as specified. Let $\Delta(a) = \text{Diam}(S_a)$. Then for every $u, v \in S_a$, there is a path uPv of length $O(\Delta(a))$ from u to v in $G(t)$ contained in S_a, and thus consisting of vertices w of degree $d(w,t) \geq t^a$. Thus each edge (x,y) of this path has resistance $1/(d(x)d(y))^b \leq 1/t^{2ab}$, so

$$R_{\mathsf{eff}}(u,v) \leq \text{Diam}(S_a) \cdot t^{-2ab}. \qquad (5.7)$$

The Matthews bound (4.4), Lemma 5.1, and the bound (5.7) give the (expected) seeded cover time $C^{*}_{S_a} = \widetilde{O}\left(\text{Diam}(S_a) \cdot t^{1-2ab}\right)$.

Details specific to Theorem 2.2. Suppose we want to find all vertices of degree at least t^a for some $a > 0$ in $G(t) \equiv G(c,m,t)$. Recall that $G(t)$ is generated by a process of attaching v_t to $G(t-1)$. At what steps were the vertices $v \in S_a$ added to $G(t)$? The expected degree of v at step t is given by (2.2), i.e., $\mathbb{E}\ d(v,t) = (1 + o(1))m(t/v)^{\eta}$. This function is monotone decreasing with increasing v. Let σ be given by

$$t^a = \left(\frac{t}{\sigma}\right)^{\eta}, \quad \text{or equivalently,} \quad \sigma = t^{1-a/\eta}. \qquad (5.8)$$

Let $s = \sigma \cdot \log^{2/\eta+1} t$. Then using Lemma 3.1, we see that all vertices added at steps $w \geq s$ have $d(w,t) = o(t^a)$ whp. On the other hand, using Lemma 3.1 again, we obtain that all vertices v added at steps $1, \ldots, s$ have degree $d(v,t) \geq (t/s)^{\eta(1-\epsilon)}$.

For Theorem 2.2, let $\Delta(a) = \text{Diam}(G(s))$, where s is as defined above. Because $\text{Diam}(G(s)) = O(\log s)$ (see (3.2)), we know that for every $u, v \in S_a$, there is a path uPv of length $O(\log t)$ from u to v in $G(t)$ contained in $G(s)$, and thus consisting of vertices w of degree

$$d(w,t) \geq \left(\frac{t}{s}\right)^{\eta(1-\epsilon)} = \frac{t^{a(1-\epsilon)}}{\text{polylog}(t)}.$$

Thus each edge (x,y) of this path has resistance

$$\frac{1}{d(x)d(y)}^{b} \leq \frac{\text{polylog}(t)}{t^{2ab(1-\epsilon)}},$$

so

$$R_{\mathsf{eff}}(u,v) \leq t^{-2ab(1-\epsilon)} \cdot \text{polylog}(t). \tag{5.9}$$

The Matthews bound (4.4), Lemma 5.1, and the bound (5.9) give the (expected) seeded cover time $C^*_{S_a} = \widetilde{O}\left(t^{1-2ab(1-\epsilon)}\right)$.

In both cases, apply the Markov inequality ($\Pr(X > A \cdot \mathbb{E}\ X) \leq 1/A$) with $\mathbb{E}\ X = C^*_{S_a}$ and $A = \log t$ to give a result whp that all vertices of degree at least t^a can be found in time

$$T_{\text{pp}}(a) = \widetilde{O}\left(\text{Diam}(S_a) \cdot t^{1-2ab}\right)$$

in pseudopreferential graphs, and in time

$$T_{\text{wg}}(a) = \widetilde{O}\left(t^{1-2ba(1-\epsilon)}\right)$$

in web graphs.

For preferential attachment graphs, we have $\eta = 1/2$, and (5.6) gives $b = 2/3$, and the time $T_{PA}(a)$ is

$$T_{\text{pa}} = \widetilde{O}\left(t^{1-(4/3)a(1-\epsilon)}\right).$$

Finally, we establish the cover time of the graph $G(t)$. This is done using (4.4) with $S = V(t)$, the vertex set of $G(t)$, i.e.,

$$C_{V(t)} \leq \max_{u,v \in V(t)} H(u,v) \log t. \tag{5.10}$$

We bound $H(u,v)$ by (4.3) as usual. The resistance $r(e)$ of every edge $e = \{x,y\}$ is

$$r(e) = \frac{1}{(d(x)d(y))^b} \leq \frac{1}{m^{2b}} = O(1).$$

Let the diameter of $G(t)$ be $\text{Diam}(G)$, which is specified for Theorem 2.1 and is $O(\log t)$ (whp) for Theorem 2.2. Thus $R_{\mathsf{eff}}(u,v) = O(\text{Diam}(G))$, since the effective resistance between u and v is at most the resistance of a shortest path between u and v. This and the bound $w(G) = \widetilde{O}(t)$ proven in Lemma 5.1 give $K(u,v) = \widetilde{O}(t \cdot \text{Diam}(G))$. Thus the cover time of the graph $G(t)$ is $\widetilde{O}(t \cdot \text{Diam}(G))$.

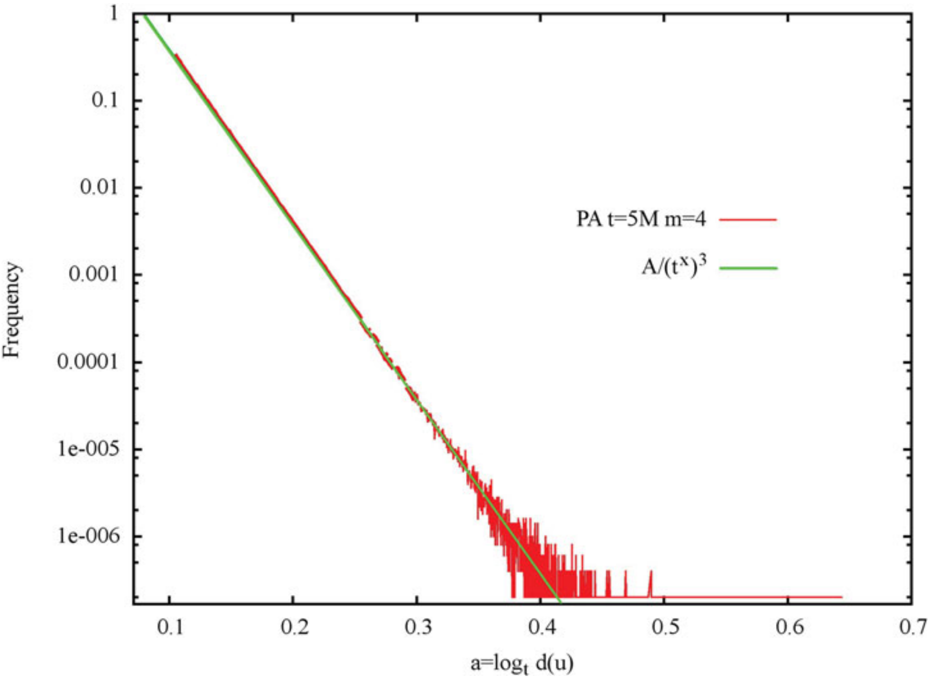

Figure 1. Degree distribution of a realization of $G(c, m, t)$, $c = 3$, $m = 4$, $t = 5 \times 10^6$.

6. Experimental Results

Theorem 2.2 gives an encouraging upper bound of order around $t^{1-(4/3)a}$ for a biased random walk to cover all vertices of degree at least t^a in the t-vertex preferential attachment graph $G(3, m, t)$. Our experiments, summarized in Figure 2, suggest that the actual bound is stronger than this. The experiments were made on $G(m, t)$ with $m = 4$ and $t = 5 \times 10^6$ vertices. The degree distribution of this graph is given in Figure 1, with both axes in logarithmic scale. More precisely, the x-axis gives the exponent a in the degree $d = t^a$, i.e., $x = \log d/ \log t$, while the y-axis represents the frequency of the vertices of degree t^a.

In Figure 2, plot SRW shows the average cover time $\tau(a)$ of all vertices of degree at least t^a by a simple random walk (the uniform transition probabilities). Plot WRW shows the average cover times by a biased random walk with $b = 1/2$ and $b = 2/3$. The plots are an average of nine runs (each) of the random walks. Both axes are in logarithmic scale. The y-axis displays $y = (\log \tau(a))/ \log t$. There are

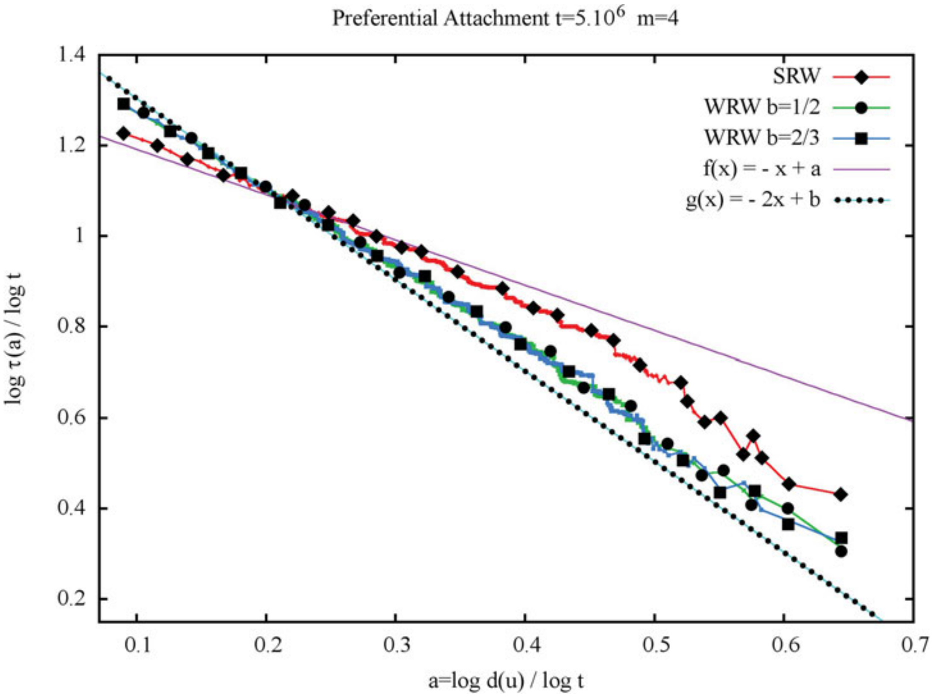

Figure 2. Cover time of all vertices of degree at least t^a in $G(c, m, t)$ as a function of a.

also three reference lines drawn in Figure 2. These lines have slopes $-a$ and $-2a$ and are included for visual inspection only. To calculate the speedup, given $x = a$, read off the $y(a)$-values y_S, y_W. The speedup is $t^{y_S - y_W}$, where $t = 5 \times 10^6$. In the upper tail, the weighted walk is about 10 times faster. Curiously, the improvement does not seem sensitive to the precise value of b.

The cover time C_G of a simple random walk on $G(m, t)$ is known and has value $C_G \sim (2m/(m-1))t \log t$; see [Cooper and Frieze 07]. The intercept of the y-axis predicted by this is $y_C = \log C_G / \log t$, which when $m = 4$ and $t = 5 \times 10^6$ is $y_C = 1.29$. This agrees well with the experimental intercept of 1.24 and helps confirm the accuracy of our simulations.

Our experimental results for Theorem 2.1 are less clear-cut, but still encouraging. Figure 3 gives the degree distribution of the underlying graph of the World Wide Web on $t = 8.7 \times 10^5$ vertices obtained from the SNAP database.[2] The

[2] Available at http://snap.stanford.edu/data/web-Google.html.

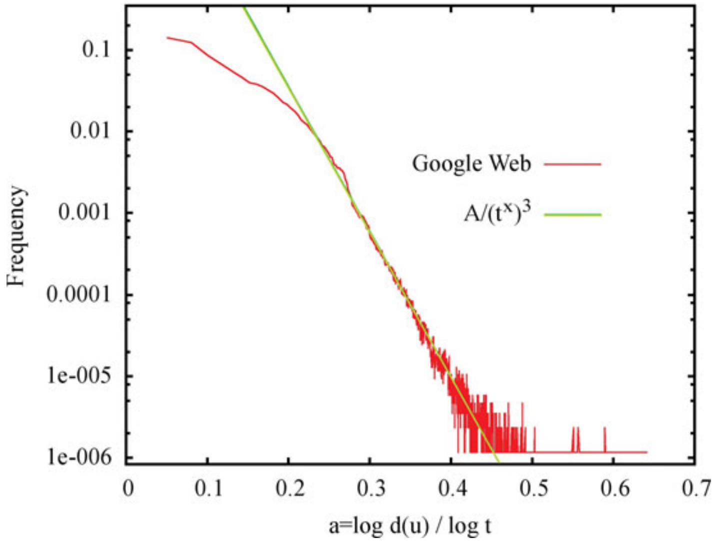

Figure 3. Degree distribution of sample of size 8.7×10^5 of G_W, the underlying graph of the World Wide Web.

power-law exponent is approximately $c = 3$, and it was crawled using a weight of $b = 2/3$. Figure 4 shows the results obtained by averaging 25 runs of the simple and weighted random walks. The weighted walk is generally about four times faster for $a > 0.43$.

Before discussing Figure 2 in greater detail, we remark that it broadly confirms the implications from our theoretical analysis: for random preferential attachment graphs, biased random walks discover quickly all higher-degree vertices while not increasing by much the cover time of the whole graph. For example, by checking the exact cover times, we observed that the biased random walk with $b = 1/2$ took on average 2.7 times longer than a simple random walk to cover the whole graph $G(3, 10^7)$, but it discovered the 100 highest-degree vertices 10 times faster than a simple random walk.

For a weighted random walk, the stationary distribution $\pi(v)$ of vertex v is given by

$$\pi(v) = \frac{1}{w(G)} \sum_{x \in N(v)} w(v, x),$$

where $w(G)$ is the sum of the edge weights of G, each edge counted twice. Thus for a simple random walk on $G(m, t)$, $\pi_S(v) = d(v)/2mt$. For the weighted random walk of Theorem 2.1 ($\eta = 1/2$ and $b = 1/2$ for $G(m, t)$), we have the following

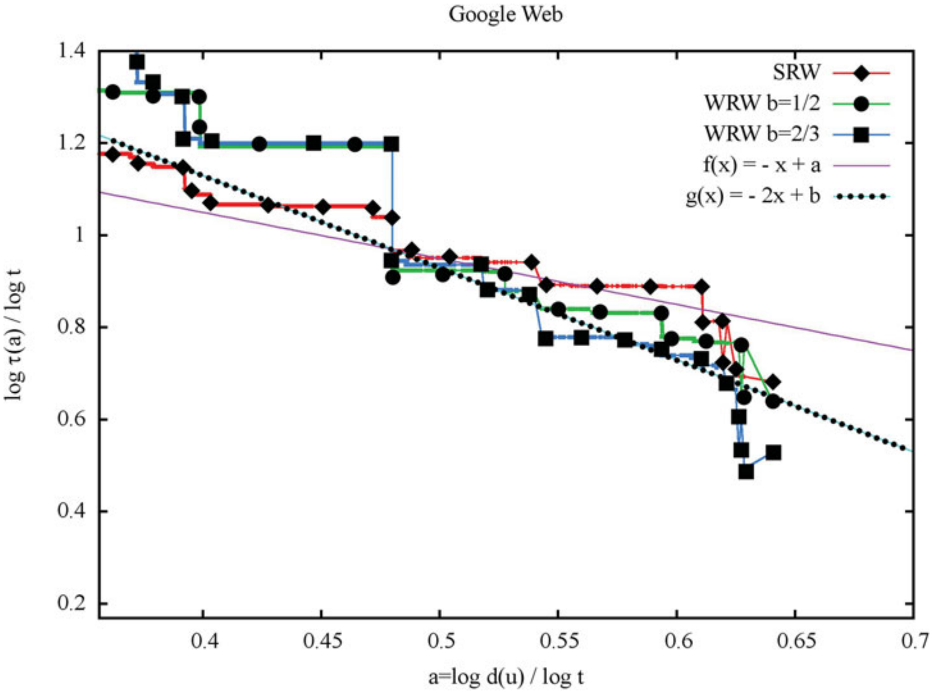

Figure 4. Cover time of all vertices of degree at least t^a in G_W as a function of a.

lower bound:

$$\pi_W(v) = \Omega\left(\frac{(d(v))^{3/2}}{t \text{ polylog}(t)}\right).$$

This bound holds because we know from Lemma 5.1 that $w(G) = \widetilde{O}(t)$ and

$$\sum_{x \in N(v)} w(v,x) = \sum_{x \in N(v)} (d(v)d(x))^{1/2} \geq (d(v))^{1/2} \sum_{x \in N(v)} (m)^{1/2} \geq (d(v))^{3/2}.$$

We can give an informal explanation of Figure 2 as follows. In the long run, the number of visits to vertex v in T steps approaches $T\pi(v)$, so the first visit to v should be at about $T(v) = 1/\pi(v)$. Since $\pi(v)$ increases with increasing degree $d(v)$, it follows that if $h > a$, we should expect to see all vertices of degree t^h before all vertices of degree t^a.

For a simple random walk, let v be a vertex of degree t^a. Then

$$T(v) \approx \frac{1}{\pi_S(v)} = \frac{2mt}{t^a} \approx t^{1-a},$$

So the SRW plot in Figure 2 should have slope $-a$, and this is indeed the case.

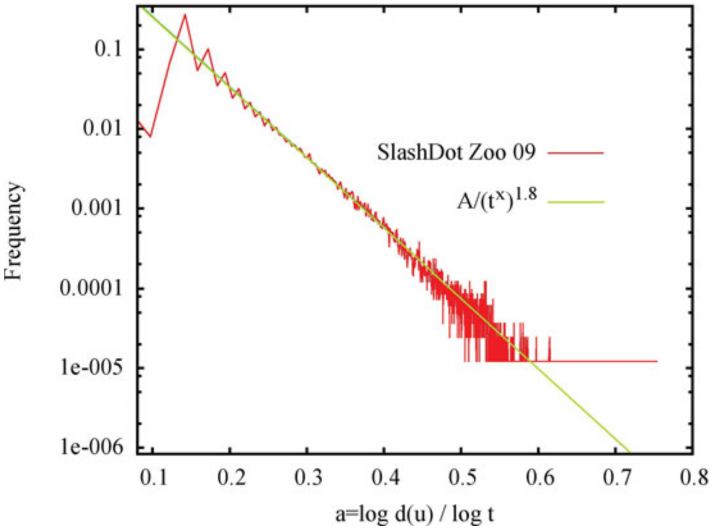

Figure 5. Degree distribution of underlying graph of a snapshot of the SlashDot network.

For a weighted random walk, the same argument gives

$$\frac{1}{\pi_W(v)} = O\left(\frac{t\log^5 t}{(t^a)^{3/2}}\right) = \widetilde{O}\left(t^{1-3/2a}\right),$$

which explains the slope of about $-3a/2$ for (at least part of) the WRW plot.

The total number $n(a)$ of vertices of degree at least t^a is approximated by $\sigma = t^{1-a/\eta} = t^{1-2a}$, where the value of σ from (5.8) is the expected step at which a vertex of degree t^a is added, and $\eta = 1/2$ for preferential attachment. Since no walk-based process can visit σ vertices in fewer than σ steps, this explains our inclusion of the line with slope $-2a$ in Figure 2.

7. Finding High-Degree Vertices in General Heavy-Tailed Graphs

In this section, we give a discussion to support a general heuristic that chooses a value of $b = 1/2$ for a weighted random walk to find high-degree vertices in heavy-tailed graphs with power law $c \geq 3$.

The setting is a graph G for which the general properties (number of edges, vertices, degree sequence, etc.) are unknown, but which we suppose to be a “small-world network” in the sense that it is connected with power-law degree

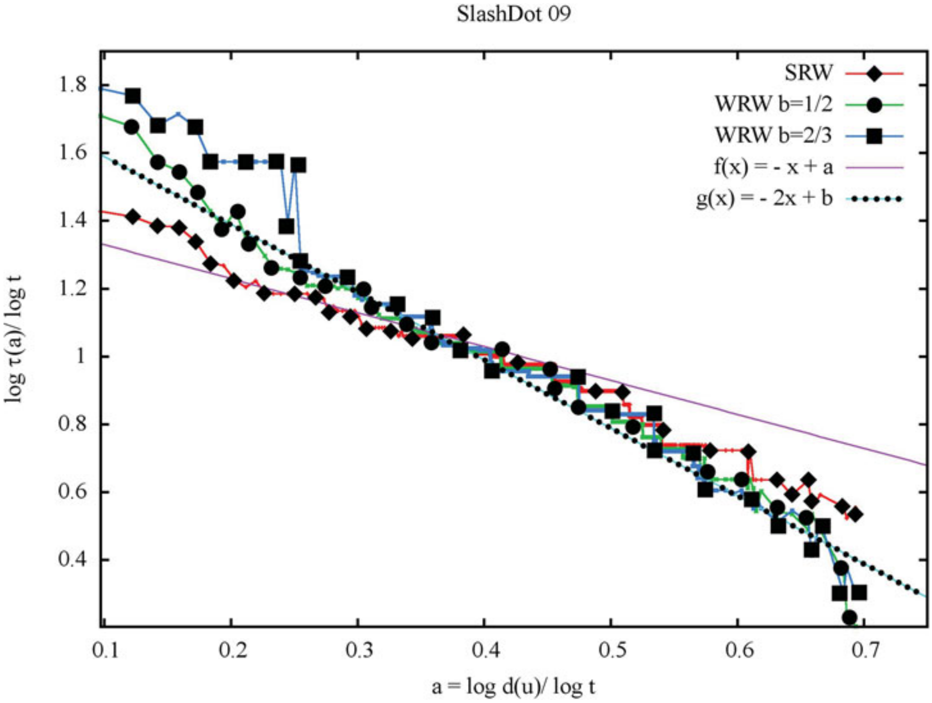

Figure 6. Cover time of all vertices of degree at least t^a in the SlashDot network as a function of a.

sequence, and the number of edges is linear in the number of vertices. We suppose as before that we want to find all high-degree vertices of G as quickly as possible. Theorems 2.1 and 2.2 do not provide any insight into the best value of b to choose, even if, as in Theorem 2.2, we assume that the power-law parameter c is greater than or equal to 3. For $c \geq 3$ at least, there is theoretical evidence to support the choice of $b = 1/2$, as we now explain.

The basic assumptions we make are as follows:

(i) We regard the power-law parameter c, the maximum degree Δ, and the number of vertices n as unknowns that cannot be used as inputs to the choice of b.

(ii) We assume that G has a power-law parameter c and heavy-tailed degree sequence $(n_1, n_2, \ldots, n_\Delta)$, where $n_k \propto nk^{-c}$.

(iii) We assume the number of edges to be $m = an$ for some constant $a > 1$.

We remark that for graphs with a heavy-tailed degree sequence, the value of Δ needs to be determined independently of c. In many papers on heavy-tailed degree sequences, an arbitrary assumption is made that $\Delta = \widetilde{O}(n^{1/2})$. In the web-graph model of [Cooper and Frieze 03], the whp value of $\Delta = \widetilde{O}(n^{1/(c-1)})$ is a property of the model.

An upper bound on the time taken by a reversible random walk to hit a vertex v from arbitrary starting position can be obtained from the following lemma. See, e.g., [Cooper et al. 12] for a proof of this lemma.

Lemma 7.1. *Let W be a lazy random walk with second eigenvalue $\lambda = \lambda(W)$ and stationary distribution π. With high probability, all vertices v with stationary distribution $\pi_v \geq \rho$ can be found in $T(\rho)$ steps, where*

$$T(\rho) = O\left(\frac{\log n}{\rho\,(1-\lambda)}\right). \tag{7.1}$$

For a weighted random walk W, the stationary distribution of vertex v is given by

$$\pi_v(W) = \frac{\sum_{u\sim v} w(u,v)}{\sum_{x\in V}\sum_{y\sim x} w(x,y)} = \frac{w_v}{w_G},$$

where $x \sim y$ means that the edge (x,y) exists.

In order to bias the walk toward high-degree vertices, we consider weights $(d(x)d(y))^b$ for edges (x,y), where $b > 0$. From the Cauchy–Schwarz inequality, we have

$$w_G = \sum_{x\in V}\sum_{y\sim x}(d(x)d(y))^b \leq \sum_{x\in V}(d(x))^{2b+1},$$

and since $d(u) \geq 1$,

$$w_v = \sum_{u\sim v}(d(u)d(v))^b \geq (d(v))^{b+1}.$$

Thus

$$\pi_v(W) = \frac{w_v}{w_G} \geq \frac{(d(v))^{b+1}}{\sum_{x\in V}(d(x))^{2b+1}}. \tag{7.2}$$

The equivalent simple random walk S has stationary distribution

$$\pi_v(S) = \frac{d(v)}{2m}.$$

For a weighted walk to reach high-degree vertices faster than a simple random walk, we would need to choose a value b for the edge weights in such a way that

$$\pi_v(W) \gg \pi_v(S) = \frac{d(v)}{2m}. \tag{7.3}$$

If we do this, and provided that $\lambda(W)$ is not too different from $\lambda(S)$, then by Lemma 7.1, the weighted walk improves the hitting time of such vertices in comparison to a simple random walk.

Let $\phi(b) = \sum_{x \in V} (d(x))^{2b+1}$. Using (7.2) and the assumption that $m = an$, $a > 1$, we can perform a worst-case analysis, replacing the requirement (7.3) by

$$\pi_v(W) = \frac{w_v}{w_G} \geq \frac{(d(v))^{b+1}}{\phi(b)} \gg \frac{d(v)}{2m} = \frac{d(v)}{2an},$$

which is equivalent to

$$(d(v))^b \gg \frac{\phi(b)}{n}. \tag{7.4}$$

Under the heavy-tailed-degree-sequence assumption, the sum $\phi(b)$ in the denominator of (7.2) can be approximated by

$$\phi(b) \propto \begin{cases} n, & c > 2b+2, \\ n \log n, & c = 2b+2, \\ n\Delta^{2b+2-c}, & c < 2b+2. \end{cases}$$

Thus we can rewrite (7.4) as

$$(d(v))^b = \Omega\left(\log n \max\left(1, \Delta^{2b+2-c}\right)\right).$$

At this point it becomes clear that there is no perfect answer. For given c, we need to choose values of b satisfying $c \geq 2b+2$.

The results of Theorem 2.2 apply when $c \geq 3$. To generalize these results for heuristic use, we assume $c \geq 3$. The value $b = 1/2$ satisfies our worst-case analysis and seems the best choice for that range. We suggest its use as an informed heuristic.

Interestingly, the value $b = 1/2$ works quite well experimentally, not only on graphs with parameter $c = 3$ (see Figures 2 and 4); it also marginally improves on simple random-walk graphs with much lower values of c, for example a SNAP[3] copy of the SlashDot network from 2009 (82 168 vertices and 948 464 edges), which has power-law parameter $c = 1.8$ (see Figures 5 and 6). Perhaps the moral

[3] Available at http://snap.stanford.edu/data/index.html. Data retrieved 12/15/2011.

is that every positive weight is good, since it biases the walk toward high-degree vertices.

When we change from a simple random walk S to a weighted walk W, we change the second eigenvalue $\lambda(W) = \lambda_2$ of the transition matrix, and hence the mixing rate. Although small-world networks are regarded as having small diameter, which means that they must be expanders and rapidly mixing for simple random walks, it is difficult to estimate the change in mixing rate for general power-law graphs.

Some bounds on the change in eigenvalue gap can be obtained from the direct comparison lemma for weighted random walks [Aldous and Fill 95, Lemma 29, Chapter 3]. The relaxation time τ_2 of a random walk is given by $\tau_2 = 1/(1 - \lambda_2)$.

Lemma 7.2. [Aldous and Fill 95] *Let (w_e) and (w_e^*) be edge weights on a simple graph G, let (w_v) and (w_v^*) be vertex weights, and let τ_2, τ_2^* be the relaxation times for the associated random walks. Then*

$$\frac{\min_e(w_e/w_e^*)}{\max_v(w_v/w_v^*)} \leq \frac{\tau_2}{\tau_2^*} \leq \frac{\max_v(w_v/w_v^*)}{\min_e(w_e/w_e^*)}.$$

The bounds on the change in mixing rate for the weighted walk we obtain in this way are a function of Δ. Since we do not know Δ, it is difficult to quantify further.

8. Conclusions

Our theoretical analysis of the number of steps required by biased random walks to discover all high-degree vertices in random t-vertex preferential attachment graphs was restricted to power laws $c \geq 3$. This was a consequence of the proof technique. The statement of Theorem 2.2 can be used to give a value of b for the walk weight for any $c > 2$, but we do not have a proof of correctness for $c < 3$. A method that provably worked down to $c = 2$ or even lower would be desirable, especially since power laws around $c = 2$ are common in real networks [Broder et al. 00].

As a heuristic, the method we propose seems to work well with weight $b = 1/2$ in many cases in which there is no performance guarantee. An example of this is the SlashDot network, which has a power law of $c = 1.6$; see Figures 5 and 6. We guess that the performance of our method depends more on the existence of a dense subgraph between high-degree vertices than on the exact power law of the degree sequence. The existence of such a dense subgraph is important to the discussion in Section 4, since it means that the effective resistance between high-degree vertices is small, thus ensuring a low cover time in (4.4).

There is a lack of agreement among models as to the range of values for high-degree vertices for a given power law c. To give an example, the web-graph model of [Cooper and Frieze 03] predicts vertices of degree $\Theta(t^{1/(c-1)})$ on t-vertex graphs, but some authors arbitrarily truncate heavy-tailed degree sequences at $O(t^{1/2})$. Any technique to find such vertices would have to be tuned to exploit the related structures in the graph.

References

[Achlioptas et al. 09] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. "On the Bias of Traceroute Sampling: Or, Power-Law Degree Distributions in Regular Graphs." *J. ACM*, 56 (2009), 21:1–21:28.

[Aldous and Fill 95] D. Aldous and J. A. Fill. "Reversible Markov Chains and Random Walks on Graphs." Available online (stat-www.berkeley.edu/pub/users/aldous/RWG/book.html), 1995.

[Baeza-Yates et al. 05] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez. "Crawling a Country: Better Strategies Than Breadth-First for Web Page Ordering." In *Proc. 14th International Conference on World Wide Web*, pp. 864–872. ACM Press, 2005.

[Barabási and Albert 99] A. Barabási and R. Albert. "Emergence of Scaling in Random Networks." *Science* 286 (1999), 509–512.

[Barabási et al. 00] A. Barabási, R. Albert, and H. Jeong. "Scale-Free Characteristics of Random Networks: The Topology of the World-Wide Web." *Physica A* 281 (2000), 69–77.

[Bollobás and Riordan 04] B. Bollobás and O. Riordan. "The Diameter of a Scale-Free Random Graph." *Combinatorica* 24 (2004), 5–34.

[Bollobás et al. 01] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. "The Degree Sequence of a Scale-Free Random Graph Process." *Random Structures and Algorithms* 18 (2001), 279–290.

[Brautbar and Kearns 10] M. Brautbar and M. Kearns. "Local Algorithms for Finding Interesting Individuals in Large Networks." In *Proceedings of ICS 2010*, pp. 188–199, 2010.

[Broder et al. 00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph Structure in the Web: Experiments and Models." In *Proceedings of the Ninth International World-Wide Web Conference WWW9*, Amsterdam 2000.

[Cooper 06] C. Cooper. "The Age Specific Degree Distribution of Web-Graphs." *Combinatorics Probability and Computing* 15 (2006), 637–661.

[Cooper and Frieze 03] C. Cooper and A. Frieze. "A General Model of Web Graphs." *Random Structures and Algorithms* 22 (2003), 311–335.

[Cooper and Frieze 07] C. Cooper and A. Frieze. "The Cover Time of the Preferential Attachment Graphs." *Journal of Combinatorial Theory B* 97 (2007), 269–290.

[Cooper et al. 12] C. Cooper, R. Elsässer, H. Ono, and T. Radzik. "Coalescing Random Walks and Voting on Graphs." In *PODC 2012 (ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing 2012)* pp. 47–56, 2012.

[Flaxman and Vera 07] A. D. Flaxman and J. Vera. "Bias Reduction in Traceroute Sampling: Towards a More Accurate Map of the Internet." In *Proceedings of WAW 2007*, pp. 1–15, 2007.

[Gjoka et al. 09] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. "A Walk in Facebook: Uniform Sampling of Users in Online Social Networks." *CoRR*, abs/0906.0060, 2009.

[Ikeda et al. 03] S. Ikeda, I. Kubo, N. Okumoto, and M. Yamashita. "Impact of Local Topological Information on Random Walks on Finite Graphs." In *Proceedings of ICALP 2003*, pp. 1054–1067, 2003.

[Lovász 96] L. Lovász. "Random Walks on Graphs: A Survey." *Bolyai Society Mathematical Studies* 2 (1996), 353–397.

[Stutzbach et al. 06] D. Stutzbach, R. Rejaie, N.G. Duffield, S. Sen, and W. Willinger. "On Unbiased Sampling for Unstructured Peer-to-Peer Networks." In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC 2006*, pp. 27–40, 2006.

Colin Cooper, Department of Informatics, King's College London, Strand, London WC2R 2LS, UK (colin.cooper@kcl.ac.uk)

Tomasz Radzik, Department of Informatics, King's College London, Strand, London WC2R 2LS, UK (tomasz.radzik@kcl.ac.uk)

Yiannis Siantos, Department of Informatics, King's College London, Strand, London WC2R 2LS, UK (yiannis.siantos@kcl.ac.uk)