Internet Mathematics Vol. 3, No. 2: 129-146

# Finding (Short) Paths in Social Networks

André Allavena, Anirban Dasgupta, John Hopcroft, and Ravi Kumar

**Abstract.** While several analytic models aim to explain the existence of short paths in social networks such as the web, relatively few address the problem of efficiently finding them, especially in a decentralized manner. Since developing purely decentralized search algorithms in general social-network models appears hard, we relax the notion of decentralized search by allowing the option of storing a *small* amount of preprocessed information about the network. We show that one can identify a small set of vertices in an undirected social network so that connectivity information of the vertices in this set can be used in conjunction with the local connectivity properties to perform decentralized search and find short paths between vertices. Our results are for random graphs with power law degree distribution generated by a variant of the expected degree model.

## 1. Introduction

In a now-famous social experiment, Stanley Milgram and his coworkers demonstrated in the 1960s that any pair of people is likely to be separated by a short chain of acquaintances [Milgram 67, Travers and Milgram 69, Korte and Milgram 78]. Each trial in their experiment had the following flavor. A source person, say Alice, in a city was given a letter to be delivered to a target person, say Bob, in a different city. Alice and Bob did not know each other but Alice would be told some basic information about Bob. Alice would be instructed to send the letter to one of her acquaintances, with the ultimate goal that the letter should reach Bob as efficaciously as possible; Alice would pass on the letter and the same set of instructions to the acquaintance she chooses for the task. The trial is successful if the letter reaches Bob. It turned out that the average number


1542-7951/06 $0.50 per page

of intermediate acquaintances in a successful trial was under six. This so-called "six degrees of separation" phenomenon has prompted serious attention from the scientific community. Very recently, the above experiment was repeated using emails and it was found that the average number of acquaintances was around 4.1 [Dodds et al. 03].

The existence of short paths has been documented in many social networks including the World Wide Web; for the web, it is a direct consequence of its small average distance [Albert et al. 99, Broder et al. 00]. It is not surprising that many social networks have short paths or small diameter. What is more intriguing is people's ability to navigate effectively in such social networks in a decentralized fashion[1] using just the local connectivity information. A lot of analytic models have been proposed to understand and explain the existence of short paths in social networks—see the surveys [Bollobás and Riordan 03, Albert and Barabasi 02, Newman 00]. While much of the work focus on *why* short paths exist between two vertices in the social network, relatively few address *how* they can be found efficiently, especially in a decentralized manner; in other words, which networks support efficient search? Watts and Strogatz [Watts and Strogatz 98] proposed a social network model that consists of a $p$-dimensional lattice superimposed with long-range links of the form $(u, v)$ where $v$ is chosen uniformly at random; they showed that these networks have short paths. Kleinberg [Kleinberg 00a, Kleinberg 00b] showed that, in general, there is no efficient decentralized search algorithm (that runs in polylogarithmic time) for these networks; however, if $v$ is chosen with probability proportional to $d(u, v)^{-p}$, where $d(u, v)$ is the lattice distance between $u$ and $v$, then (and only then) an efficient decentralized search algorithm exists. Kleinberg [Kleinberg 01] also considered a more general hierarchical model and decentralized search with partial information about the underlying network. See the survey by Kleinberg [Kleinberg 06].

In this paper we focus on the algorithmic aspects of a search in random graphs generated by a variant of the expected degree model. While this model is mathematically tractable, a general-purpose decentralized search algorithm for this model seems unlikely. Therefore, we relax the requirements of an efficient decentralized search algorithm and consider the option of storing a *small* amount of preprocessed information about the network that will, in conjunction with the local information, let us find paths between two vertices. We demand that the preprocessed information be small relative to size of the network, that the

[1]Online services such as LinkedIn (http://www.linkedin.com), Friendster (http://www.friendster.com), Spoke (http://www.spokesoftware.com), and Orkut (http://www.orkut.com) are attempting to exploit this phenomenon. These services help users to expand their social circle and link to other users with overlapping interests; users typically accomplish this by local navigation, say, by looking at friends of friends.

search algorithm uses only this and the local connectivity information, and that the path found by the search algorithm is of the order of the diameter of the network.

### 1.1. Main Contributions

We show that one can identify a small set of vertices (the *core*) in an undirected social network so that connectivity information about the vertices in this set, along with local information about degree and connectivity, can be used to perform an efficient decentralized search. Our results are obtained for the random graphs with power law degree distributions generated by a variant of the expected degree model. For a power law distribution with parameter $\beta \in (2, 3)$, the size of the core is roughly $\tilde{O}(n^{1-2/\beta})$. The length of paths found by our algorithm is $O(\ln n)$ and matches the diameter of such graphs [Chung and Lu 02, Chung and Lu 03].

The intuition behind our algorithm is very simple. The algorithm consists of a preprocessing stage and a querying stage. In the preprocessing stage, we designate a set of vertices of sufficiently high degree in the graph to be the core. We explicitly compute all-pair shortest paths for the vertices in the core and store the information in a table $T$. In the querying stage, we are given two vertices $s$ and $t$ and the goal is to find a path between $s$ and $t$ using only local information and the table $T$. The key step is to find a short path from both $s$ and $t$ to the core via a mixture of random and deterministic walks. We start with a random walk from $s$ until a vertex of sufficiently high degree is reached, at which point we switch to a deterministic walk by following the neighbor with the highest degree until a vertex in the core is reached; call this vertex $s_c$. We perform a similar operation by starting from $t$ until $t_c$, a vertex in the core, is reached. Note that all the operations in this step are decentralized and can be performed just using local connectivity and degree information. Now, finding a path between $s_c$ and $t_c$ is easy using the table $T$. Note that even if table $T$ were not computed, performing a random walk from $s_c$ and $t_c$, but restricting the walk to only visiting vertices that are in the core, will eventually establish a path between $s_c$ and $t_c$.

Our result is the first of its kind for graph models that are significantly more general than the Watts–Strogatz family of random graphs. We believe our algorithm, owing to its simplicity, will find applications in peer-to-peer networks and distributed settings, where decentralized search is often desirable, and in the context of developing algorithms for massive graphs, where it is not possible to look at the entire graph to answer path queries on the fly; for a sample setting, see [Faloutsos et al. 04].

### 1.2. Related Work

There have been several generative models proposed for social networks such as web graphs [Aiello et al. 00, Barabasi and Albert 99, Aiello et al. 02, Kumar et al. 00, Cooper and Frieze 03, Bollobás and Riordan 04]. A pervading theme in many of these models is that new edges do not point to randomly chosen vertices, but rather to vertices chosen proportional to their "popularity," i.e., their current degree. It was also shown that these graphs have small average distance and diameter [Bollobás and Riordan 04, Lu 01, Chung and Lu 02, Chung and Lu 03]. Fabrikant, Koutsoupias, and Papadimitriou [Fabrikant et al. 02] proposed the FKP model of vertices on a plane in which edges are added to optimize the trade-off between distances and "centrality" in the graph. It is important to note that the graphs generated by neither the Watts–Strogatz model nor the FKP model have the power law degree distribution that is often observed in many social networks. To amend this, Chung and Lu [Chung and Lu 04] proposed a hybrid power law graph model that incorporates power law degree distribution while retaining small-world properties; they did not consider search problems in the networks.

Chung and Lu [Chung and Lu 02, Chung and Lu 03] analyzed a simple variant of the configuration model and showed that there is a core of size $n^{1-c/\ln\ln n}$ such that almost all vertices are within distance $\ln\ln n$ of the core. Though their definition of core is similar to ours, their work differs from ours in two aspects. Firstly, their focus was more on the existence of short paths and not on obtaining a search algorithm that finds these short paths. Secondly, the core they obtain is substantially bigger than ours. Mihail, Saberi, and Tetali [Mihail et al. 06] show that in power law random graphs, the expected rate at which a random walk with lookahead discovers the nodes of the graph is sublinear.

The problem of searching a social network from an experimental point of view was considered by Adamic et al. [Adamic and Adar 03, Adamic et al. 01] and Kim et al. [Kim et al. 02]. They explore the path-finding strategy of following the neighbor with the highest degree; their results, however, are from an experimental point of view.

## 2. Model

A number of random graph models have been proposed for social networks with skewed degree distribution. Notable among these are the expected degree model, in which each edge is chosen independently according a certain probability, and the configuration model, in which the set of edges is generated as a random

matching between the vertices. Most of these models generate graphs with a unique giant component, along with a number of components whose sizes are asymptotically smaller. Note that the problem of finding a path between two vertices is interesting only when both the vertices are in the giant component. To address this technicality, we adopt a modified version of the expected degree model, detailed in Coja-Oghlan and Lanka [Coja-Oghlan and Lanka 06], in order to generate a random graph with a single connected component.

**Notation.** For a graph $G$, let $V_G$ denote the set of vertices and $E_G$ denote the set of edges. For a vertex $v \in V_G$, let $d_G(v)$ denote the degree of a vertex $v$ in $G$, and for a subset $V \subseteq V_G \setminus \{v\}$, let $e_G(v, V_G)$ denote the number of edges $(v, w) \in E_G, w \in V$. For $V \subseteq V_G$, let $\mathrm{vol}_G(V) = \sum_{v \in V} d_G(v)$.

Let $\mathbf{d} = d_1, \ldots, d_n$ be a given sequence of degrees, where $d_v$ represents the degree of a vertex $v$. Let $d_{\min} > 3$ (respectively $d_{\max}$) be the minimum (respectively maximum) degree in the sequence. Given this degree sequence, the random graph $G = G(\mathbf{d})$ on the vertex set $[n]$ is defined as follows. Define the $n \times n$ probability matrix $P$ entry-wise as $P_{uv} = d_u d_v / \sum_w d_w$, and generate a graph $G$ by independently rounding to 0/1 (modulo symmetry) each of the entries of the probability matrix $P$. Note that the expected degree of $v$ in $G$ is $\mathbf{E}[d_G(v)] = d_v$. We are interested in random graphs with a power law degree sequence, and therefore consider only the case when the sequence $\mathbf{d}$ is distributed according to a power law with exponent $\beta \in (2, 3)$, i.e., $|\{v \mid d_v = i\}| = n_0/i^\beta$, where the normalizing factor $n_0 = n / \sum_{i=d_{\min}}^{d_{\max}} i^{-\beta}$. Also, note that $d_{\max} = n_0^{1/\beta}$. The sum of degrees, denoted $2m$, is then equal to

$$2m = \sum_{i=d_{\min}}^{d_{\max}} \frac{n_0}{i^\beta} \cdot i = \Theta(n).$$

Finally, we apply the following procedure on the resulting graph $G$ to define a subset of vertices $H$. Let $c_0$ be any constant such that $3 < c_0 < d_{\min}$.

1. Discard all vertices in $V_G$ that are not in the giant component of $G$.
2. Let $V_H = V_G \setminus \{v \mid d_G(v) \leq 0.01 \cdot d_{\min}\}$.
3. While there is a vertex $v \in V_H$ that has at least

$$\max\{c_0, \exp(-d_{\min}/c_0) d_G(v)\}$$

neighbors in $G \setminus H$, remove $v$ from $V_H$.

Let $H$ be the graph in $G$ induced by the vertices in $V_H$. Denote the degree of each vertex in $H$ to be $d_H(v)$. The following set of results are from [Coja-Oghlan and Lanka 06]. The first claim states that $H$ constitutes a large fraction of the vertices and of the total degree of $G$. The second claim states that, for vertices of degree at least $\ln n$, the degrees in $G$ and $H$ are both close to the expected degree. Lastly, the main claim proves that $H$ has a large spectral gap.

**Theorem 2.1.** [Coja-Oghlan and Lanka 06] *The graph $H$ obtained from $G$ as a result of the iterative steps above satisfies the following properties with high probability over the generation process of the random graph:*

*(a) $H$ is connected, and the total sum of degrees of $H$ is close to the total expected degree of $G$, i.e., $\sum_{v\in H} d_v \geq (1-\exp(-100d_{\min}/c_0))n$.*

*(b) For every vertex $v$ with expected degree $d_v > \ln n$, the actual degree $d_G(v)$ satisfies*

$$d_v - 2\sqrt{d_v \ln n} \leq d_G(v) \leq d_v + 2\sqrt{d_v \ln n}.$$

*Furthermore, the degree of $v$ in $H$ satisfies*

$$d_H(v) \geq d_G(v) - \max\{c_0, \exp(-d_{\min}/c_0)d_G(v)\}.$$

*(c) Finally, let $Q$ be the transition matrix associated with a random walk in the graph induced by $H$. Then, the spectral gap of the matrix $Q$ is at least $1 - c_0/\sqrt{d_{\min}}$.*

Henceforth, we assume that all the statements in Theorem 2.1 are true. For vertices with expected degree $d_v > \ln n$, we also denote the closeness of the quantities $d_v$, $d_G(v)$, and $d_H(v)$ by simply writing $d_G(v) = \Theta(d_v)$ and $d_H(v) = \Theta(d_v)$.

Given any two vertices $s$ and $t$, both of which are in $V_H$, our goal is to find a path between $s$ and $t$. For any degree $d$, define the set $S_d$ of vertices to be the set of vertices of expected degree at least $d$. For any set $X$ of vertices we will denote the total expected degree as $\mathrm{vol}(X) = \sum_{v\in X} d_v$.

## 3. The Algorithm

Define the *core* to be the set of vertices $\mathcal{X}$ whose actual degree $d_H(\cdot)$ is at least $d_{\max}(1 - \frac{\ln n}{d_{\max}^{3-\beta}})$. From Theorem 2.1, we have the following.

**Proposition 3.1.** *The expected degree of each vertex in the core is* $\Theta(d_H(v))$*, and the size of the core* $\mathcal{X}$ *is at most* $|\mathcal{X}| = O(n^{(1-2/\beta)} \ln n)$.

The formal description of the algorithm appears in Algorithm 1.

For each of the Steps 1–3, we need to show that we succeed in at most $O(\ln n)$ steps. First, we show that the Step 1 of the algorithm succeeds.

**Lemma 3.2.** *In Step 1, the algorithm sees at most* $O(\ln^{\beta} n)$ *vertices and, with probability at least* $1 - 1/n$ *over the random walk choices, finds a path of length* $O(\ln n)$ *to a vertex of expected degree at least* $\ln n$.

**Proof.** Since the random walk is performed on the graph $H$, in steady state the probability of the walk being in any set $S$ is equal to the total degree $\mathrm{vol}_H(S)$ of the vertices in the set $S$. Now, let $S = \{v \in H \mid d_v \geq \ln n\}$. Thus, the total expected degree of $S$, $\mathrm{vol}(S)$, is at least

$$\begin{aligned}
\mathrm{vol}(S) &= \sum_{v:d_v \geq \ln n} d_v = \sum_{i=\ln n}^{d_{\max}} \frac{n_0}{i^{\beta}} \cdot i \\
&\geq \int_{\ln n}^{d_{\max}} \frac{n}{(i-1)^{\beta-1}} di \\
&= \frac{n_0}{(\beta-2)} \left( \ln^{2-\beta} n - d_{\max}^{2-\beta} \right) \\
&\geq \frac{n_0 \ln^{2-\beta} n}{2\beta(\beta-2)}.
\end{aligned}$$

**Algorithm 1**

Given a query for a path between vertices $s$ and $t$ in $V_H$, do the following:

1. Start with the given vertex $s$ and do a random walk for $\ln n$ steps. If at any point the random walk hits a vertex of degree at least $\ln n$, go to Step 2. If no vertex of degree at least $\ln n$ was encountered even after $\ln n$ steps of the random walk, abort this walk and repeat this step all over.
2. After reaching a vertex of degree $\ln n$, continue to take the maximum degree neighbor deterministically till the walk reaches one of the vertices in the core, say $s_c \in \mathcal{X}$.
3. Do the same for finding a path between $t$ and a core vertex $t_c$.
4. Establish a path between $s_c$ and $t_c$ using $\mathcal{X}$.

By Theorem 2.1(b), the total degree $\text{vol}_G(S)$ and simultaneously $\text{vol}_H(S)$ are at least $\Omega\left(\frac{n_0 \ln^{2-\beta} n}{2\beta(\beta-2)}\right)$, and the total volume of $H$ is also $\Theta(\text{vol}(V_G))$. Thus, the total probability assigned to $S$ by the stationary distribution is at least $\Omega\left(\frac{n_0 \ln^{2-\beta} n}{4m\beta(\beta-2)}\right)$, i.e., $\Omega(\frac{1}{\ln^{\beta-2} n})$. Now, by Theorem 2.1(c), the number of steps required to bring down the state-probability vector to a point-wise distance of at most $1/n$ from the stationary probability distribution is given by

$$O\left(\frac{1}{1-c_0/\sqrt{d_{\min}}} \ln(\frac{n\text{vol}_H(S)}{d_{\min}})\right) \leq O(\ln n).$$

So, after each short walk of length $O(\ln(n))$ from the start node $s$, the probability of being in the set $S$ is at least $\Omega(\frac{1}{\ln^{\beta-2} n})$. Hence, if we take $\Theta(\ln^{\beta-1} n)$ independent random walks, each of length $\Omega(\ln n)$, then the expected number of times that the random walk ends up in $S$ is $\Theta(\ln^{\beta-1} n \cdot \ln^{2-\beta} n) = \Theta(\ln n)$. Thus, with probability $1 - 1/n$, at least one of the random walks will actually end up in $S$. □

Next, we show that the walk in the second step of the algorithm does not terminate at any local maxima other than the ones that have been stored during preprocessing. This is the crux of our algorithm.

**Lemma 3.3.** *In Step 2 of the algorithm, if we start from a vertex of expected degree at least* $2 \ln n$*, with probability* $1 - o(1)$ *(over the random graph space) we reach a vertex in* $\mathcal{X}$ *in* $O(\ln n)$ *steps.*

**Proof.** We will evaluate the probability of increasing the degree at each step. More specifically, we evaluate the probability of not having a neighbor of expected degree at least $d$ when at a vertex $u$ of expected degree $d_u = k$. Namely, we will show that if the expected degree of a vertex is $k$, then with high probability it will have a neighbor whose expected degree is $d$; the value of the target $d$ will depend on the specific case with which we are dealing. Since Step 2 of the algorithm works only with vertices of degree greater than $\ln n$, by applying the union bound, it is easy to see that this statement implies that each vertex of degree $k$ will have a neighbor $v$ such that $d_H(v) = \Omega(d)$. The expected number of edges between

the vertex $u$ and the subset $S_d$ is given by $\mathbf{E}[e_H(u, S_d)] = \frac{d_u \mathrm{vol}(S_d)}{2m}$. Thus,

$$\begin{aligned}\mathbf{E}[e_H(u, S_d)] &= d_u \sum_{i=d}^{d_{\max}} \frac{i \cdot n_0 i^{-\beta}}{2m} \\ &= \frac{d_u n_0}{2m} \int_d^{d_{\max}} i^{1-\beta} di \\ &= \gamma d_u \left( \frac{1}{d^{\beta-2}} - \frac{1}{d_{\max}^{\beta-2}} \right),\end{aligned}$$

where the constant $\gamma$ is defined as

$$\gamma = \frac{n_0}{2m(\beta - 2)}.$$

We now break the rest of the analysis into two cases. The first is when our target degree is $d \leq \mu d_{\max}$ for some constant $\mu$, and the second one is when the target degree $d \geq \mu d_{\max}$. For a vertex $v$, define $N(v)$ to be its set of neighbors. We show that the probability of success at each stage is $1 - \Omega(\frac{1}{n})$, and then we can take the union bound over the $O(\ln n)$ steps.

**Case 1.** For a vertex with current degree $d_u$ and target degree $d \leq \mu d_{\max}$, where $\mu < \frac{1}{2}$ is some constant, we show that we can find a neighbor of degree at least $d = d_u^{\frac{2}{\beta-1}}$. We will just need to show that the probability of failing to find such a high expected degree neighbor is small. Since the total number of edges formed in $H$, i.e., $e_H(u, S_d)$ is a sum of random variables, the result is a simple application of the following version of the Chernoff bound. Suppose that $X_1, \ldots, X_n$ are independent random variables in $[0, 1]$, each with expectation $\mathbf{E}[X_i] = \mu_i$; then,

$$\Pr\left[ \sum_i X_i < \sum_i \mu_i - t \right] < \exp\left( \frac{-t^2}{4 \sum_i \mu_i} \right).$$

So now, since $d \leq \mu d_{\max}$, we have

$$\begin{aligned}\mathbf{E}[e_H(u, S_d)] &= \gamma d_u \left( \frac{1}{d^{\beta-2}} - \frac{1}{d_{\max}^{\beta-2}} \right) \\ &\geq \frac{\gamma d_u}{d^{\beta-2}} \left(1 - \mu^{\beta-2}\right).\end{aligned}$$

Since $d = d_u^{\frac{2}{\beta-1}}$,

$$\mathbf{E}[e_H(u, S_d)] = \Omega\left( d_u d_u^{-\frac{2\beta-4}{\beta-1}} \right) = \Omega\left( d_u^{\frac{3-\beta}{\beta-1}} \right).$$

Thus, applying the Chernoff bound with $t = \sqrt{\mathbf{E}[e_H(u, S_d)]}\ln(d_u)$, we see that there must be nonzero edges between $u$ and $S_d$ in $H$. Since none of these edges are lost to the construction of $H$, we have proved the presence of edges between $u$ and $S_d$ in the graph $H$ too. Thus, from a vertex of expected degree $d_u$, with high probability we can walk to a vertex of expected degree at least $d_u^{\frac{2}{\beta-1}}$ in one step. Hence, we can reach a vertex of degree $\mu d_{\max}$ in at most $O(\ln\ln n)$ steps.

**Case 2.** In this case, we analyze the walk over the higher degree vertices, when the current expected degree $d_u$ is at least $\mu d_{\max}$. We will show that in at most two steps we reach the core. Define

$$\lambda = \frac{4\ln n}{\mu(\beta-2)\gamma d_{\max}^{3-\beta}}.$$

Let $d = (1-\lambda)d_{\max}$. We show that we can reach a vertex of degree $d$ from a vertex of degree $\mu d_{\max}$. Revisiting the expected number of edges $\mathbf{E}[e_H(u, S_d)]$, we have

$$\begin{aligned}
\mathbf{E}[e_H(u, S_d)] &= \gamma d_u \left(\frac{1}{d^{\beta-2}} - \frac{1}{d_{\max}^{\beta-2}}\right) \\
&\geq \frac{\gamma d_u}{d_{\max}^{\beta-2}} \left(\frac{1-(1-\lambda)^{\beta-2}}{(1-\lambda)^{\beta-2}}\right) \\
&\geq \frac{\gamma d_u}{d_{\max}^{\beta-2}} \cdot \frac{(\beta-2)\lambda}{(1-\lambda)^{\beta-2}} \\
&\geq (\beta-2)\gamma\lambda\ d_u d_{\max}^{2-\beta} \\
&\geq \frac{4d_u \ln n}{\mu d_{\max}} \geq 4\ln n.
\end{aligned}$$

Thus, the actual number of edges $e_H(u, S_d)$ is again at least $2\ln n$ with probability at least $1 - \frac{1}{n^2}$, and hence we are able to reach a vertex of degree at least $(1-\lambda)d_{\max}$. Applying a similar argument, we can prove that from a vertex of degree at least $(1-\lambda)d_{\max}$, we can reach the core.

Combining the results of the two cases, the proof is complete. □

**Lemma 3.4.** *$\mathcal{X}$ has diameter $O(\ln n)$.*

**Proof.** We use the well-known fact that a random graph $G(n,p)$ has diameter $O(\ln n)$ when $p = \ln n/n$ and show that the probability of having any edge within the core is at least $\ln n/n$.

Let us consider two nodes $u$ and $v$ from $\mathcal{X}$ and the probability that there exists an edge between them. By the definition of the core,

$$\begin{aligned}\Pr[e_H(u,v)] = \frac{d_u d_v}{2m} &\geq \frac{d_{\max}^2}{2m}\left(1 - \frac{\ln n}{d_{\max}^{3-\beta}}\right)^2 \\ &= \Theta(n^{2/\beta-1}) \\ &= \Theta\left(\frac{\ln n}{|\mathcal{X}|}\right),\end{aligned}$$

where the last equality follows from Proposition 3.1. □

Thus, we obtain our main theorem.

**Theorem 3.5.** *The algorithm stores information about a core of size $O(n^{1-\frac{2}{\beta}} \ln n)$ nodes and can find a path from any vertex to another by looking at at most $O(\ln^{\beta} n)$ nodes in all. The length of the path found between any two vertices is at most $O(\ln n)$, with probability $1 - o(1)$.*

**Proof.** The size of the core is $O\left(n^{1-\frac{2}{\beta}} \ln n\right)$. The algorithm needs to store only the core and a sparse set of edges connecting the vertices of the core. The number of vertices looked at and the length of the path is obtained by adding the estimates obtained in Lemmas 3.2, 3.3, and 3.4. □

Note that, as part of the preprocessing, we can compute all-pair shortest paths for the vertices in $\mathcal{X}$. This will let us compute the path between $s_c$ and $t_c$ rather easily via a table lookup. Alternately, if we do not wish to store this information explicitly, we could continue to do random walks from $s_c$ and from $t_c$ till they hit a common vertex, restricting the random walk to hit only vertices in $\mathcal{X}$. It can also be shown that, with high probability, after $\sqrt{|\mathcal{X}|}$ steps the walks will collide, establishing a path between $s_c$ and $t_c$. Thus, Step 4 is taken care of.

*Special case of $\beta = 2$.* This is really a special case because the average number of edges per vertex is not constant anymore but logarithmic in the number of vertices: $m = \Theta(n \ln n)$. This changes the derivation of some of the proofs, but not the results themselves.

However, it is worth noticing that in case of $\beta = 2$, the size of the core is polylogarithmic in $n$. Moreover, the processing phase is no longer necessary. A random walk within the core will succeed in linking two nodes in a polylogarithmic number of steps.

## 4. Discussion

### 4.1. Other Models

We now discuss the extension of our algorithms to other models such as the preferential attachment/copying model [Aiello et al. 00, Barabasi and Albert 99, Aiello et al. 02, Kumar et al. 00, Cooper and Frieze 03, Bollobás and Riordan 04]. Since these graphs evolve over time, we need to define the core differently: the core is the set of the oldest nodes.

To make our discussion simpler, let us assume the following model. We start with an initial set of nodes $I$. At each time step $t$ we add a node $u$—in fact let's label it $t$—and $k$ edges from $t$ to the previous nodes. Those end points can be chosen with preferential attachment, for example, or uniformly at random.

It is obvious that in this simple model all the edges point to older nodes. If one knows the age of the nodes, it would be extremely easy to reach the core: just follow whichever edge brings us the furthest back. For now, let us assume that we know the age of the nodes. The question is how long does it take to reach the core by following edges that point to older nodes. If we are at a node $t$, it is straightforward to see that we should have an edge to a node with expected label $t/k$; this assumes that end points are chosen uniformly at random. If we instead use some preferential attachment, or variant of it (copying for example), the likelihood of picking small labels would be even higher. Thus, it would take $O(\ln t)$ steps to get to the core.

### 4.2. The Directed Case

In this section we discuss the possibility of extending our algorithm to the directed case. An analog of our algorithm might be hard in the directed case without any assumptions on the correlation between the in-degrees and out-degrees. Suppose that we define the core to be the set of high out-degree vertices. While it becomes easy to reach any vertex from the core, the ease of reaching the core itself from any vertex is not obvious. Similarly, if we define the core to be the set of high in-degree vertices, then the converse problem happens. Note that the above two sets might have poor connectivity among them if no correlation between in-degrees and out-degrees exists.

We therefore assume that the in-degrees and out-degrees are correlated. Under this assumption, we define the core to be the set of nodes of high out-degree. We drop the requirement of finding a path between two vertices and instead focus on an algorithm to reach the core from any vertex in a decentralized fashion.

The model is a simple generalization of the undirected case. We assume that the out-degree sequence is a power law with exponent $\beta \in (2, 3)$. The in-degree

sequence can be arbitrary but should be consistent with the definition of correlation that we describe below. The main difference from the undirected case is that the probability that an edge exists between $u$ and $v$ is now proportional to the product of the out-degree $o_u$ of $u$ and the in-degree $i_v$ of $v$, i.e.,

$$\Pr[e(u \to v)] \propto o_u i_v.$$

**4.2.1. Correlation.** One way to define correlation between the in-degree and the out-degree sequences is to say that if a vertex has out-degree $x$, then it has in-degree at least $x^\alpha$, where $\alpha$ is a parameter. Note that $\alpha = 1$ corresponds to perfect correlation, i.e., the undirected case and $\alpha = 0$ corresponds to no correlation at all, i.e., the in-degree of a vertex does not reveal any information about the out-degree of a node.

**4.2.2. $\alpha$-correlation.** In fact, we will significantly weaken the correlation assumption so that we only need to impose the above condition in an aggregate sense for a large set of vertices, instead of for each vertex. Our relaxed definition of correlation is the following. Suppose that $S_d$ is the set of vertices with out-degree at least $d$. Given a parameter $\alpha \geq 0$, the in-degrees and out-degrees are said to be *$\alpha$-correlated* if for all $d$, the sum of the in-degrees of the vertices in $S_d$ is at least as large as when assuming that every node (of out-degree $x$) has in-degree $x^\alpha$, i.e.,

$$\sum_{v \in S_d} i_v \geq \sum_{i=d}^{d_{\max}} (i^\alpha \cdot \text{ number of nodes of outdegree } i) = \sum_{i=d}^{d_{\max}} i^{\alpha-\beta} n_0.$$

We show that, with an $\alpha$-correlation where $\alpha > \beta - 2$, we can reach the set of highest out-degree nodes with very high probability in $O(\ln \ln n)$ steps. We can also show a weak lower bound: if there is a weaker correlation (with $\alpha < \beta - 2$), or none, then no algorithm can reach our core faster than in $O(n^\epsilon)$ steps for some constant $\epsilon > 0$.

**Lemma 4.1.** *Suppose that the in-degree and the out-degree sequences are $\alpha$-correlated, where $\beta$ is the power-law exponent of the out-degree sequence and $\alpha > \beta - 2$. Define the core to be the set of nodes of out-degree at least $d_{\max}(1 - \frac{\lambda \ln n}{d_{\max}^{\alpha+2-\beta}})$ for some constant $\lambda > 0$. The size of the core is $O(n^{\frac{\beta-\alpha-1}{\beta}} \ln n)$. Then, starting from a vertex of out-degree at least $\ln n$, we can reach the core with very high probability in $O(\ln \ln n)$ steps with a deterministic algorithm.*

**Proof.** This proof is essentially similar to the undirected case, using $\beta - \alpha - 1$ instead of $\beta - 2$. Note that if we have a correlation of $\alpha$, then we also have a

correlation of $\alpha'$ for any $\alpha' < \alpha$. Thus, without loss of generality, we assume that $\alpha < \beta - 1$.

Let us study the probability for a node $u$ to have an edge to a node of higher out-degree. Again, let $o_u$ be the current out-degree and $d$ the minimum target out-degree.

$$\begin{aligned}
\mathbf{E}[(u, S_d) \in E_H] &= \sum_{v \in S_d} \Pr[(u, v) \in E_H] \\
&\geq o_u \sum_{i=d}^{d_{\max}} \frac{i^\alpha \cdot n_0 i^{-\beta}}{2m} && \text{(correlation property)} \\
&= o_u \gamma \left( \frac{1}{d^{\beta-\alpha-1}} - \frac{1}{d_{\max}^{\beta-\alpha-1}} \right). && \text{(with } \gamma = \tfrac{n_0}{2m(\beta-\alpha-1)}\text{)}
\end{aligned}$$

The derivation of the probability that a vertex $u$ of out-degree $o_u$ is not connected to any vertex of degree $d$ or higher is the same as in the undirected case. This time we take $d$ to be

$$d = o_u^{\frac{\beta-\alpha}{2(\beta-\alpha-1)}},$$

where the lower bound on $\alpha$ implies that $\frac{\beta-\alpha}{2(\beta-\alpha-1)} > 1$, i.e., our target degree $d$ is larger than our current out-degree $o_u$.

Applying the Chernoff bound, the probability of failure when making progress while $d = o(d_{\max})$ is then bounded by $\exp(-\gamma o_u^{\frac{\alpha-\beta+2}{2}})$. Similarly, the probability of failure in the last step when $d = \Omega(d_{\max})$ is $\Theta\left(\frac{1}{n^{(\beta-\alpha-1)\lambda\gamma\mu}}\right)$ for some appropriate constant $\mu$.

Again, as in the undirected case, we can decrease the constant $\lambda$ in the size of the core by taking an extra step. But, we cannot hope to decrease the size of the core any further since the probability of reaching a node of higher degree goes to 0. □

We now show that our assumption on correlation is tight with respect to our definition of the core.

**Lemma 4.2.** *If there exists a $d_0 = n^\varepsilon$ for some constant $\varepsilon > 0$ such that*

$$\sum_{v \in S_{d_0}} i_v \leq \sum_{i=d_0}^{d_{\max}} (i^\alpha \cdot \text{ number of nodes of out-degree } i) = \sum_{i=d_0}^{d_{\max}} i^{\alpha-\beta} n_0,$$

*for some $\alpha$ such that $\alpha < \beta - 2$, then any algorithm from a certain class of algorithms is going to need at least an expected $\Omega(n^{\varepsilon(\beta-2-\alpha)})$ steps to reach $S_{d_0}$.*

**Proof.** Note that $S_{d_0}$ is not significant compared to the whole graph: $|S_{d_0}| = n^{\kappa}$, where $\kappa = 1 - \varepsilon(\beta - 2) < 1$ .

The class of algorithms we study is the following: the algorithm knows of a list $L_t$ of nodes that it has previously visited, with $t = |L_t|$, and knows of the out-degree of the nodes in $L_t$, but not of the nodes not in $L_t$. At each round $t$, it either picks a node $v$ uniformly at random from all possible nodes or picks one out-going edge from some node in $L_t$; let us call $v$ the vertex to which that edge points. Vertex $v$ is added to $L_t$ to form $L_{t+1}$, and the neighbors of $v$ and their out-degrees become known to the algorithm. Note that the algorithm that we constructed certainly falls in this class. We are in fact giving "extra power" to the algorithm as we allow it to backtrack for free, or to randomly start over.

The probability of succeeding in any step is the maximum of the probability of success of the two alternatives of the algorithm. Let us start by examining the probability of success when following an edge of a node that we already know.

Starting from some node $u$ not in the core with out-degree $o_u$, the probability of reaching a node $v$ in $S_{d_0}$ is at most

$$\begin{aligned}
\Pr[\exists v \in S_{d_0} \mid (u,v) \in E_G] &\leq \sum_{v \in S_{d_0}} \Pr[(u,v) \in E_G] \\
&\leq o_u \sum_{i=d_0}^{d_{\max}} \frac{i^{\alpha-\beta} n_0}{2m} && \text{(correlation hypothesis)} \\
&\leq o_u \frac{n_0}{2m} \int_{d_0-1}^{d_{\max}} i^{\alpha-\beta} di \\
&\leq o_u \frac{n_0}{2m(\beta-\alpha-1)} \frac{1}{d_0^{\beta-\alpha-1}} \\
&\leq \frac{o_u \gamma}{d_0^{\beta-\alpha-1}} && \text{(with } \gamma = \tfrac{n_0}{2m(\beta-\alpha-1)} (\tfrac{d_0}{d_0-1})^{\beta-\alpha-2}) \\
&\leq \frac{\gamma}{(d_0)^{\beta-\alpha-2}} && \text{(since } o_u \leq d_0) \\
&= \frac{\gamma}{n^{\varepsilon(\beta-\alpha-2)}}. && \text{(since } d_0 = n^{\varepsilon})
\end{aligned}$$

On the other hand, the probability of succeeding by randomly jumping to a new node and by chance selecting a node in $S_{d_0}$ is even smaller:

$$\begin{aligned}
\Pr[\text{success by randomly jumping to a node}] &= n^{\kappa-1} \\
&= n^{-\varepsilon(\beta-2)} \\
&\ll \gamma n^{-\varepsilon(\beta-2-\alpha)}.
\end{aligned}$$

The probability of success is the maximum of the two previously detailed probabilities, namely,

$$p = \Pr[\text{success in step } t] \leq \max(n^{-\varepsilon(\beta-2)}, \frac{\gamma}{n^{\varepsilon(\beta-2-\alpha)}}) = \frac{\gamma}{n^{\varepsilon(\beta-2-\alpha)}}.$$

The expected time to succeed is $T = 1/p$, hence the claimed result. □

**Acknowledgments.** We thank Jon Kleinberg for many useful discussions. Part of the work for this paper was done by the first and second authors when they were at Cornell University and by the fourth author while he was at IBM Almaden Research Center and visiting Cornell University.

## References

[Adamic and Adar 03] L. A. Adamic and E. Adar. "How to Search a Social Network." arXiv:cond-mat/0310120, 2003.

[Adamic et al. 01] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. "Search in Power-Law Networks." *Phys. Rev. E* 64 (2001), 046135.

[Aiello et al. 00] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model for Power Law Graphs." *Experiment. Math.* 10:1 (2000), 53–66.

[Aiello et al. 02] W. Aiello, F. Chung, and L. Lu. "Random Evolution in Massive Graphs." In *Handbook on Massive Data Sets*, edited by J. Abello, P. M. Pardalos, and M. G. C. Resende, pp. 97–122. Norwell, MA: Kluwer, 2002.

[Albert and Barabasi 02] R. Albert and A-L. Barabasi. "Statistical Mechanics of Complex Networks." *Reviews of Modern Physics* 74 (2002), 47.

[Albert et al. 99] R. Albert, H. Jeong, and A-L. Barabasi. "Diameter of the World Wide Web." *Nature* 401 (1999), 130–131.

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

[Bollobás and Riordan 03] B. Bollobás and O. Riordan. "Mathematical Results on Scale-Free Random Graphs". In *Handbook of Graphs and Networks*, edited by S. Bornholdt and H. G. Schuster, pp. 1–37. Berlin: Wiley–WCH, 2003.

[Bollobás and Riordan 04] B. Bollobás and O. Riordan. "The Diameter of a Scale-Free Random Graph." *Combinatorica* 24:1 (2004), 5–34.

[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." *Computer Networks* 33:1–6 (2000), 309–320.

[Chung and Lu 02] F. Chung and L. Lu. "The Average Distance in a Random Graph with Given Expected Degrees." *Proceedings of the National Academy of Sciences* 99:25 (2002), 15879–15882.

[Chung and Lu 03] F. Chung and L. Lu. "The Average Distance in a Random Graph with Given Expected Degrees." *Internet Mathematics* 1:1 (2003), 91–114.

[Chung and Lu 04] F. Chung and L. Lu. "The Small World Phenomenon in Hybrid Power Law Graphs." In *Complex Networks*, edited by E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, pp. 91–106, Lecture Notes in Physics 650. Berlin: Springer, 2004.

[Coja-Oghlan and Lanka 06] A. Coja-Oghlan and A. Lanka. "The Spectral Gap of Random Graphs with Given Expected Degrees." To appear in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings*, Lecture Notes in Computer Science 4051. Berlin: Springer, 2006.

[Cooper and Frieze 03] C. Cooper and A. Frieze. "A General Model of Web Graphs." *Random Structures and Algorithms* 22:3 (2003), 311–335.

[Dodds et al. 03] P. S. Dodds, R. M. Lukose, and D. J. Watts. "An Experimental Study of Search in Global Social Networks." *Science* 301 (2003), 827–829.

[Fabrikant et al. 02] A. Fabrikant, E. Koutsoupias, and C. Papadimitriou. "Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet." In *Automata, Languages and Programming: 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8–13, 2002, Proceedings*, edited by P. Widmayer et al., pp. 110–122, Lecture Notes in Computer Science 2380. Berlin: Springer, 2006.

[Faloutsos et al. 04] C. Faloutsos, K. S. McCurley, and A. Tomkins. "Fast Discovery of Connection Subgraphs." In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 118–127. New York: ACM Press, 2004.

[Kim et al. 02] B. J. Kim, C. N. Yoon, S. K. Han, and H. Jeong. "Path Finding Strategies in Scale-Free Networks." arXiv:cond-mat/0111232, 2002.

[Kleinberg 00a] J. Kleinberg. "Navigation in a Small World." *Nature* 406 (2000), 845.

[Kleinberg 00b] J. Kleinberg. "The Small-World Phenomenon: An Algorithmic Perspective." In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 163–170. New York: ACM Press, 2000.

[Kleinberg 01] J. Kleinberg. "Small-World Phenomenon and the Dynamics of Information." In *Advances in Neural Information Processing Systems*, vol. 14. Cambridge, MA: MIT Press, 2001.

[Kleinberg 06] J. Kleinberg. "Complex Networks and Decentralized Search Algorithms." To appear in *Proceedings of the International Congress of Mathematicians.* Zurich: European Mathematical Society Publishing House, 2006.

[Korte and Milgram 78] C. Korte and S. Milgram. "Acquaintance Networks Between Racial Groups: Application of the Small World Method." *Journal of Personality and Social Psychology* 15 (1978), 101.

[Kumar et al. 00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. "Stochastic Models for the Web Graph." In *Proceedings of 41st Annual Symposium on Foundations of Computer Science*, pp. 57–65. Los Alamitos, CA: IEEE Press, 2000.

[Lu 01] L. Lu. "The Diameter of Random Massive Graphs." In *Proceedings of the Twelfth Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 912–921. Philadelphia: SIAM, 2001.

[Mihail et al. 06] M. Mihail, A. Saberi, and P. Tetali. "Random Walks with Lookahead in Power Law Random Graphs." To appear in *Internet Mathematics*, 2006.

[Milgram 67] S. Milgram. "The Small-World Problem." *Psychology Today* 1 (1967), 62–67.

[Newman 00] M. Newman. "Models of the Small World." *Journal of Statistical Physics* 101 (2000), 819–841.

[Travers and Milgram 69] J. Travers and S. Milgram. "An Experimental Study of the Small World Phenomenon." *Sociometry* 32 (1969), 425.

[Watts and Strogatz 98] D. J. Watts and S. H. Strogatz. "Collective Dynamics of Small-World Networks." *Nature* 393 (1998), 440–442.

André Allavena, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (aallavena@cs.uwaterloo.ca)

Anirban Dasgupta, Yahoo! Research, 701 First Ave., Sunnyvale, CA 94089 (anirban@yahoo-inc.com)

John Hopcroft, Department of Computer Science, Cornell University, Ithaca, NY 14850 (jeh@cs.cornell.edu)

Ravi Kumar, Yahoo! Research, 701 First Ave., Sunnyvale, CA 94089 (ravikumar@yahoo-inc.com)

Received May 3, 2005; accepted September 13, 2006.