THESIS FOR THE DEGREE OF LICENTIATE OF
PHILOSOPHY

Searching in a Small World

Oskar Sandberg

CHALMERS | GÖTEBORG UNIVERSITY

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and Göteborg University
Göteborg, Sweden 2005

Searching in a Small World
Oskar Sandberg

NO 2005:52
ISSN 1652-9715

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 1000

Printed in Göteborg, Sweden 2005

Searching in a Small World

Oskar Sandberg

Department of Mathematical Sciences
Chalmers University of Technology
Göteborg University

Abstract

The small-world phenomenon, that the world's social network is tightly connected, and that any two people can be linked by a short chain of friends, has long been a subject of interest. Famously, the psychologist Stanley Milgram performed an experiment where he asked people to deliver a letter to a stranger by forwarding it to an acquaintance, who could forward it to one his acquaintances, and so on until the destination was reached. The results seemed to confirm that the small-world phenomenon is real. Recently it has been shown by Jon Kleinberg that in order to search in a network, that is to actually find the short paths in the manner of the Milgram experiment, a very special type of a graph model is needed.

In this thesis, we present two ideas about searching in the small world stemming from Kleinberg's results. In the first we study the formation of networks of this type, attempting to see why the kind of connections necessary may arise naturally. A different criterion on the network which also makes the efficient searches possible is derived, and based on it an algorithmic model is proposed for how searching can become possible as a network evolves.

In the second paper, we propose a method for searching in small-world networks even when the participants are oblivious to their own and others positions in the world. This is done by assigning nodes positions in an idealized world based on the clustering of connections between them, and then searching based on these positions. The problem is motivated by applications to computer networks, and our method is tested on real world data.

Acknowledgements

It feels premature to make acknowledgments when the real task is far from completed, but I am thankful for the continued support of my advisers Olle Häggström and Devdatt Dubhashi, as well as the kindness and understanding of my initial adviser Nanny Wermuth. Thanks to Ian Clarke in discussion with whom many of these ideas first arose. Also thanks to Ilona, Johan, Mikael, Peter, São, my officemate Viktor and all the other people at the Department of Mathematical Sciences who make spending almost all my waking time at work an eminently enjoyable way of life.

Contents

Chapter 1

Introduction

1.1 Milgram's Small-World Experiment

It has almost become cliché to start out a work on random networks with a reference to experimental psychologist Stanley Milgram's famous small-world experiment. Yet, since the experiment has been the catalyst for so much of the thought about networks that has followed it, it is impossible to not do so.

The small-world phenomenon, which had been discussed already before Milgram proposed his experiment, is based on an idea familiar to most people. It says, in a nutshell, that our social world is held together by short chains of acquaintances - that even complete strangers, though they may not have a mutual acquaintance, will be linked as friends of friends of friends through just a few steps. Most people have anecdotes to this effect, and the expression "it is a small world" has become part of everyday speech.

In order to explore this matter further, Milgram proposed a simple experiment. Starting with volunteers picked at random from a city in the American mid-west, he would give them packages intended to be forwarded to a target of his choice by mail. The rules had a catch however: recipients of the package could not send it to just anybody, nor just directly to final target, but had to send it to somebody with which they were acquainted (defined, for the experiment, as somebody with which they were on first name basis) [28]

[34].

Milgram and his associates conducted the experiment several times with different starting groups (from Wichita in Kansas, Omaha in Nebraska, and Boston) and several targets (a stockbroker in Boston, and the wife of Divinitees student at Yale). The reported results were, at first glance, a stunning confirmation that the world really is small: the successful chains found their way from person to person in a strikingly small number of steps. The number of steps that was cited as the average in one of studies, six, so caught the peoples imagination that the term "six degrees of seperation" has become part of our cultural folklore.

With time, the so called "small-world phenomenon" has reached beyond psychology and sociology. In recent years especially, much work has been done on explaining the phenomena with mathematical methods. This started with work aimed at showing that random graphs have a small diameter [6][12], continued through the celebrated small-world models of Watts and Storagtz [37][35], and, importantly for the present work, those of Jon Kleinberg [23] [24]. In recent years the field has become extremely popular, with hundreds of papers produced annually.

As noted above, Milgram's experiment is the starting point of almost all small-world discussion, and few papers come without a reference to the 1967 article (from Psychology Today, a popular magazine rather than a scientific journal) describing it. The success of the experiment, and that we really do live in a world where people can find short chains of friendships between one another, has become part of the accepted canon motivating the theoretical work.

It is worthwhile, however, to take a step back and consider what Milgram actually found. The whole story is, as always, somewhat more complicated than the popular anecdote[1]. The problem, it turns out, is that while successful chains really were short in the experiments, the number of successfully completed chains was very small. In his first study, which started with people selected through a newspaper add in Wichita, Kansas, and aimed to deliver a folder to the wife of a divinities student at Yale, only three of the sixty chains that Milgram started were successful. In the later study, starting with

people in Nebraska and Boston, the success rate varied between 24 and 35 percent: substantially better, but still hiding a large proportion of the chains. Part of the reason for the better success rates in the second round of experiments, it seems, was that Milgram went out of his way to make the parcel seem valuable: a dark blue passport with Harvard University written in gold letters on the cover.

In an article by psychologist Judith Kleinfeld [25], a large number of other problems with the study are cited. The subjects were selected in ways that made them less likely to be truly random subjects, and several replications of the experiment, where the success rate was too low to draw any conclusions, were never published. Kleinfeld also concludes that while many similar studies have been conducted (with varying success) in specialized fields and single cities, she could at the time find no large scale replication of the small-world experiment. In fact, she speculates, from a psychological perspective there may be two different small world phenomena worth studying: not only why and whether we form friendships so that we world really is small, but also why the idea that we do is so compelling to us. The latter, of course, is not a question that mathematical work can be of much assistance in answering.

Since Kleinfeld's article was written, however, a large scale replication of the small-world experiment has been carried out using the Internet. Dodds et. al. at the Columbia University small-world project[2] have solicited volunteers to start chains aimed at reaching 18 preselected targets in 13 countries [13]. Perhaps once again reflecting the popular allure of the concept, they got a very large number of volunteers. 98,847 signed up for the Columbia experiment, and of these 25 percent actually went on to start chains.

While the large number of chains seems promising, the observed success rates make those achieved by Milgram seem stellar. Of the 24,163 chains started, only 384 (a little over 1.5 percent) actually reached their intended destinations. This can be considered to sup-

[1]Readers of Swedish may, for more exposition about the experiment, turn to the chapter on the topic in [20]. Also Kleinfeld's paper [25], cited below, contains a thorough, critical, discussion of the topic.

[2]http://smallworld.columbia.edu

port Milgram's hypothesis that the success rate depends on the perceived value of the parcel: few, if any, people will see much value in an Internet chain letter. For the completed chains, the average of steps was 4.05, which again sounds good, but, as the authors of the study themselves note, must be considered misleading due to the conditioning on success.

The problem is, as should be clear to most observers, one of positive self selection. Since one would expect chains to become more likely to fail with every person they pass through, the large percentage of failures masks most of the longer chains from the average. Indeed, if every chain were to fail once it reached some fixed, small, number of steps, the conclusion that chains are short conditioned on success would mean nothing: the very fact that they succeeded implies that they were short.

Mathematically, this is a simple application of conditional probability. It holds that, if we let L be the number of steps a chain takes, and A the event that it succeeds, then

$$\mathbf{P}(L = \ell \text{ and } A) = \mathbf{P}(A \,|\, L = \ell)\mathbf{P}(L = \ell).$$

From this we can express the true probability of a chain having length ℓ as:

$$\mathbf{P}(L = \ell) = \frac{\mathbf{P}(L = \ell \text{ and } A)}{\mathbf{P}(A \,|\, L = \ell)} = \frac{\mathbf{P}(A)}{\mathbf{P}(A \,|\, L = \ell)}\mathbf{P}(L = \ell \,|\, A).$$

Since $\mathbf{P}(A \,|\, L = \ell)$ is expected to be small for large values of ℓ, we can expect these values to be underrepresented in the data compared to their true frequency.

The advantage of the Internet based experiment over previous, letter based ones, is, however, that the use of the computer network allowed the Columbia team to track the chains at every step. This allowed them to see where and at what rate queries terminated. The findings show that the number of people of who chose not to continue the chain stayed constant at around 65 percent for all steps after the first. This would seem to indicate that it is user apathy and disinterest, rather than a difficulty or frustration in carrying out the experiment, that causes attrition of the chains. Using this data,

Dodds et al let:

$$\mathbf{P}(A|L = \ell) = \prod_{i=0}^{\ell-1}(1 - r_i).$$

with r_i denoting the proportion of chains that were discontinued in each step (thus meaning that $\mathbf{P}(A|L = \ell) \approx 0.35^{\ell}$). The formula above then makes an estimate of L's true distribution possible. The upper tail the distribution is difficult to estimate due to a lack of data (none of the chains started lasted more than eleven steps), but a median value of L, based on the data, is calculated as 7. In other words, even in a similar experiment with no attrition, we should expect half the chains to complete by the seventh step.

So where does this leave the world? Most probably, the data would seem to indicate, it is indeed small, at least in many cases. But it is also a lot more complicated than a single number or experiment can explain, and the mythical "six degrees" are likely to remain just that. In the words of the Columbia team:

> Our results suggest that if individuals searching for remote targets do not have sufficient incentives to proceed, the small-world hypothesis will not appear to hold, but that even a slight increase in the incentives can render searches successful under broad conditions. More generally, the experimental approach adopted here suggests that empirically observed network structure can only be interpreted in light of the actions, strategies, and even perceptions of the individuals embedded in the network: Network structure alone is not everything.

1.2 The Mathematics of Small Worlds

Mathematical exploration of the small-world problem predates even Milgram's experiment, but development was initially slow. The problem is known to have been discussed in the sixties at MIT, leading to a paper by I. de Sola Pool and M. Kochen, but because of a lack of progress it was first published in 1978 [12]. Since then a lot of work

has been done, especially through computer simulation, but many of the theoretical questions remain open.

Seen from a mathematical perspective, the small-world phenomenon is a problem of graph theory[3]. The question explored in the experiment becomes one of measuring the distances between vertices in a graph, where the distance between two vertices is the length of the shortest path connecting them (so called geodesic distance). One wants to bound the mean such distance, or, ideally, the maximum distance between any two vertices, also known as the graphs diameter.

The most common model for random graphs, attributed, alternatively to Erdös and Renyi [16] and to Solomonoff and Rapoport [33] is taking a set V of vertices, and connecting each disjoint pair of vertices with probability p. These graphs have many interesting properties with have been source of much study in probabilistic combinatorics [4] [7] [22]. In particular, there exists p for which there is a giant connected component of size $\Omega(n)$, and the diameter does indeed scale logarithmically in size.

Such completely random graphs, however, are seldom very good models for the type of networks one finds in nature. While they have a low diameter, they do not have another important property of most observed networks: clustering. Clustering is most easily stated as the principle that two vertices that share a common neighbor are more likely to be connected than two vertices chosen at random from V. This is obviously not the case in the above model, where all vertex pairs are independently connected with the same probability.

Formally, one defines the clustering coefficient of a (random) graph as the average (expected) portion of a vertex's neighbors which are also connected to each other. Clearly $C = 1$ for a complete graph, $C = 0$ for trees, and $C = p$ for random graphs of the type

[3]The concept of a structure of points and the lines connecting them is ubiquitous is many scientific fields. It is called a "graph" in mathematics, which the points denoted as "vertices", and the lines as "edges". In computer science it is usually called a "network", which "nodes" and "links" or "connections". In physics such a structure is a "system", which has "sites" and "bonds". Finally, in sociology one usually refers to a network of people, or "actors", and contacts, friendships, or "ties".

discussed above. Non-complete graphs with higher clustering coefficients can be constructed, most easily through so called nearest-neighbor graphs.

Nearest-neighbor graphs are constructed by starting with a finite regular lattice, which in 1 dimension is just n nodes set in a line, often closed on itself (so the line becomes a cycle), and adding edges from each vertex to the k nearest in the lattice. Such a network can easily be seen (see [29]) that, if $k < \frac{2}{3}n$:

$$C = \frac{3(k-2d)}{4(k-d)}$$

where d is the dimension of the lattice.

In modern terminology, a small-world graph is one which displays both the small diameter of the random graph, and the heavy clustering of organized nearest-neighbor graphs. Of course, the terms "small diameter" and "heavy clustering" are rather ambiguous, and have often been used somewhat loosely. However, "small diameter" is usually understood to mean that the diameter should grow logarithmically (or at most polylogarithmically) in the size of the graph, while "heavy clustering" usually can be taken to mean that the clustering coefficient should not fall considerably when the graph grows but the number of edges per node stays constant.

The small-world model of Watts and Strogatz from 1998 [36] is an explicit construction of such graphs. They start with a structured, clustered graph, such as a nearest-neighbor graph, and then "re-wire" a proportion q of the edges by changing one end to a uniformly random destination. By allowing q to vary from 0 to 1, one can interpolate between a structured model and one very similar to the random graphs of Erdös and Renyi. Through computer simulation, Watts and Strogatz concluded that for a large portion of the q values, the resulting graphs would have both properties identified as characteristic of the small world.

The rewiring model is, however, rather difficult to analyze analytically. Something more approachable is provided by the subsequent model of Newman and Watts [31], where extra, random "shortcut" edges are added to an existing graph, rather than rewired from existing edges. The simplest version of this model is to start with

a k nearest-neighbor graph on a one dimensional cycle, and a random matching of the vertices to provide exactly one shortcut edge at each vertex. The shortcut adds $2k$ additional neighbor pairs of each vertex, of which we may assume few if any will be connected. Thus:

$$C \geq \frac{(k-2)}{2(k-1)}$$

where we approach equality when $n >> k$. Bollobas and Chung [8] proved already in the 1980's that a cycle with a random matching has a diameter which is with high probability $\theta(\log n)$, so certainly this graph has at most logarithmic diameter, making it a small-world network.

With this example in mind, it isn't difficult to imagine that the same thing holds for similar mixes of structured and unstructured graphs. Rigorous results about this have been relatively elusive, however, with most published results relying on simulation or so called mean field approximations. Summaries of existing results can be found in the reviews by Newman [30] [29], as well as in a draft by Durrett [14] which also attempts to collect some of the more rigorous results available.

1.3 Navigation

The small-world models of the 1990's go a long way toward illustrating the type of dynamics that should be expected from real world networks with a short diameter. They have stimulated a lot of study, and have been a triumph in the sense of managing to explain a lot of different real world networks, arrising in fields from phsyiology to sociology to physics, with simple models.

It should be noted, however, that there is a rather large gap between what can be said about these models, and what the Milgram experiment perhaps showed about the world's social network. While the combination of structured network and shortcuts can explain that there is are short paths between people, Milgram's experiment would seem to illustrate not only that these paths exist, but that people, working with very little information, can find them.

Jon Kleinberg tackled the questions in 2000. The result was a seminal paper [23] in which he showed that the previous small-world models could, in fact, not explain this fact. It is simply not possible for any algorithm, working only with local knowledge of the graph, to efficiently paths when a grid-like graph is subjected to uniformly random rewiring or the addition of uniformly distributed shortcuts. The expected number of steps to find one point from another is lower bounded by a root of the network size, and is thus exponential in the diameter.

Moving on from this, Kleinberg allowed for a wider family of (semi-) random graphs. Similarly to Newman and Watts, he starts with an underlying grid and adds shortcut edges, but Kleinberg allows the probability that two vertices are connected by a long edge to depend on the distance between them in the grid. In particular, the probability that two non-adject vertices x and y is allowed to belong to the family:

$$\frac{d(x,y)^{-\alpha}}{\Delta_\alpha} \tag{1.1}$$

where d is the distance between the vertices in the underlying grid, and Δ_α is a normalizer. In this family, he showed it is the case where α equals the dimension of the grid, and only that case, which allows for efficient navigation (finding paths from one vertex to another in a polylogarithmic number of expected steps).

1.3.1 Why Navigation Matters

Before proceeding to presenting Kleinberg's results in more detail, we diverge to discuss why these results are important. At first they may seem mostly like a mathematical curiosity - it was, after all, not people's ability to find paths between each other that Milgram set out to measure. He was interested in the original small-world problem: are we closely connected to everyone else? The quality which many people find so appealing, that we may all be linked as friends within a few steps, has little to do with algorithmic nature of how such paths are found. Similarly, many of the applications where a small graph diameter is important, such as those from epidemiology, have little to do with finding paths.

That Milgram chose to let the people taking part themselves do the searching was simply experimental necessity - nobody had global access to the worlds social network so no better way of routing was possible. This is less true today then it was then, and people have studied social networks in cases where the entire graph has been revealed [2] [26] - we do, for particular cases, in the second paper of this thesis.

The importance of the particular question of navigation has grown since Milgram did his initial work, however. Milgram's ideas were first published a year before the early ancestor of today's Internet came into existence, but since then this medium has grown into a ubiquitous and essential part of our lives. Systems like the Internet (whose name is derived from "Internetworking protocol", meaning a protocol meant to connect many smaller, clustered, networks) depend by their very nature on navigation, or, as it is commonly called, routing. Complicated addressing and router system are set up exactly solve the problem of sending packets of information between hosts in the network using efficient paths.

Given this, it is not surprising that simple probabilistic models which allow for efficient routing should be of interest. The author of this work himself first approached these problems while trying to find ways to efficiently organize peer-to-peer overlay networks (networks of users connected over the Internet) in distributed ways. The second paper presented below illustrates techniques of which combine these ideas with those about social networks, exploiting the small-world to allow routing in networks that directly connect only friends.

1.4 Kleinberg's Results

In this section we will review some of the navigability results that form the basis of the continued work covered later in the thesis. We will show, using Kleinberg's proofs from [23], that the family given by (1.1) allows for polylogarithmic routing at, and only at, one value of the α. Kleinberg originally did his work in a two dimensional setting - inspired by Milgram's experiment - but where needed we shall work with a one dimensional base grid for simplicity. Similar

arguments apply for grids of any dimension. The one dimensional situation is particularly well explored in [5], but we use Kleinberg's method rather than their abstractions below.

To start with we need to define what Kleinberg calls a decentralized routing algorithm. This means, in essence, that the routing at each vertex takes place using only locally available information, and no centralized authority with global knowledge is involved. If we let a query (message) travel through the network, we define $\{X_t\}_{0 \leq t \leq T}$ as the position of the query at step t. $Y = X_0$ is the starting point, and for the random time T, $Z = X_T$ is the destination.

Definition 1.1. *A method for selecting the next step of a query* $\{X_t\}_{0 \leq t \leq T}$ *is a* decentralized algorithm *if, the choice of* X_{t+1} *depends only on:*

1. *The coordinate system and connections of the underlying grid structure.*
2. *The coordinates in the grid of the target* z*.*
3. *The coordinates in the grid of* X_j *and all* X_j*'s neighbors, for* $0 \leq j \leq t$*.*

While the concept in some ways characterizes algorithms which work locally (as people do when forwarding messages to friends) it is a little misleading to think of these as local routing algorithms. For one thing the last criteria is looser, allowing one to use the entire history of the query[4], and secondly the knowledge about the grid and coordinate system is in some ways global. Decentralized routing when no knowledge about the positions is given is a problem we tackle in the second paper of this thesis.

We now let, as stated, the underlying grid be a closed directed cycle of n vertices. For simplicity, we will also move from un-directed graphs to directed, and we assume the cycle consists of directed clockwise edges. Distance with respect this grid is the circular distant along the direction of the links. Each vertex chooses the destination of an additional directed edge, which we henceforth refer to as a

[4]This strengthens the result, since it is not needed for the upper bounds presented, but the lower bounds hold in spite of it.

shortcut, independently with probability given by (1.1) for some α[5]. We let $\mathcal{A}$ denote a decentralized algorithm, and $\tau_{\mathcal{A}} = E_{\mathcal{A}}(T)$ be the expected number of steps it takes to find the destination under this algorithm.

Theorem 1.2. *For any decentralized algorithm $\mathcal{A}$:*

- $\tau_{\mathcal{A}} \geq k_1(\alpha)n^{(1-\alpha)/2}$ *if* $0 \leq \alpha < 1$.
- $\tau_{\mathcal{A}} \geq k_2(\alpha)n^{(\alpha-1)/\alpha}$ *if* $\alpha > 1$.

where k_1 and k_2 depend on α but not on n.

This, of course, leaves out the critical case where $\alpha = 1$, which we discuss below. It is very much a case of the first condition leading to too few shorter shortcuts, and the second leading to too few longer shortcuts, which is exactly what the method of proof will be in each case.

Proof. **The case $0 \leq \alpha < 1$:** First we note that in this case, we can lower bound Δ_α by

$$\begin{aligned} \sum_{i=1}^{n-1} x^{-\alpha} &\geq \int_1^{n-1} x^{-\alpha} && (1.2) \\ &= (1-\alpha)^{-1}((n-1)^{1-\alpha} - 1) && (1.3) \\ &\geq \rho n^{1-\alpha} && (1.4) \end{aligned}$$

for some constant ρ depending on α but not n.

Now we let U be the set of nodes from which the target z is within distance n^δ where $\delta = (1-\alpha)/2$. Of course, $|U| \leq n^\delta$.

Now define an event, A, as the event that within λn^δ steps, with $\lambda = \rho/4$, the message reaches a node whose shortcut leads to a node within U. The probability of any particular shortcut existing is $\leq 1/\Delta_\alpha$, so if we let A_i denote the event of finding such a shortcut in the i-th step, then

$$\mathbf{P}(A_i) \leq \frac{|U|}{\Delta_\alpha} \leq \frac{n^\delta}{\rho n^{1-\alpha}}.$$

[5] All the results discussed below hold also when there is more than one shortcut, and when the cycle is a k nearest-neighbor graph. Only the values of the constants differ.

Since $A = \bigcup_{i \leq \lambda n^\delta} A_i$ it follows that

$$\begin{aligned} \mathbf{P}(A) &\leq \sum_{i \leq \lambda n^\delta} \mathbf{P}(A_i) \\ &\leq \frac{\lambda n^{2\delta}}{\rho n^{1-\alpha}} \\ &= \frac{1}{4} \end{aligned}$$

Now we let B be the event that distance from the starting point to the target, $d(Y, Z) > n/2$. Since we are choosing starting points uniformly, this gives

$$\mathbf{P}(B) \geq \frac{1}{2}.$$

Since $\mathbf{P}(A^c) > 3/4$, elementary probability tells us that:

$$\mathbf{P}(A^c \cap B) \geq \frac{1}{4}.$$

Now consider the T, the number of steps until we reach our target. The event $T \leq \lambda n^\delta$ cannot occur is $A^c \cap B$ does, since in order to reach the target in less then λn^δ steps, we must at some point before then find a shortcut ending in U.

$$\mathbf{P}(T \leq \lambda n^\delta \mid A^c \cap B) = 0 \Rightarrow E_{\mathcal{A}}(T \mid A \cap B) \geq \lambda n^\delta.$$

And by restriction it then holds that

$$\begin{aligned} \tau_{\mathcal{A}} &= E_{\mathcal{A}}(T) \\ &= E_{\mathcal{A}}(T \mid A^c \cap B)\mathbf{P}(A^c \cap B) \\ &= \frac{1}{4}\lambda n^\delta. \end{aligned}$$

A suitable choice of $k_1(\alpha)$ now gives the result.

The case $\alpha < 1$: We start by bounding the probability that a node u has a shortcut destination v that is more than m steps away. Let

$\epsilon = \alpha - 1 > 0$

$$\begin{aligned}
\mathbf{P}(d(u,v) > m) &\leq \sum_{j=m+1}^{N-1} j^{-\alpha} \\
&\leq \int_m^\infty x^{-\alpha} dx \\
&= \epsilon^{-1} m^{-\epsilon}
\end{aligned}$$

Now let $\gamma = 1/(1+\epsilon)$, and A_i be the event that in the i-th step, we find a shortcut longer than n^γ. Also let $\mu = \min(\epsilon, 2)/4$, and

$$A = \bigcup_{i \leq \mu n^{\epsilon\gamma}} A_i$$

be the event that we find such a shortcut in the first $\mu n^{\epsilon\gamma}$ steps. Now

$$\begin{aligned}
\mathbf{P}(A) &\leq \sum_{i \leq \mu n^{\epsilon\gamma}} \mathbf{P}(A_i) \\
&\leq \mu n^{\epsilon\gamma} \epsilon^{-1} (n^{-\gamma})^\epsilon \\
&= \mu \epsilon^{-1} \leq \frac{1}{4}.
\end{aligned}$$

Similarly to the first case, we let B be the event that $d(Y,Z) > n/2$, which means that $P(A^c \cap B) > 1/4$. If $A^c \cap B$ occurs, then $T \geq \mu n^{\gamma\epsilon}$, because the total distance moved in the first $\mu n^{\gamma\epsilon}$ steps is $\leq \mu n^{\epsilon\gamma+\gamma} = \mu n < n/2$. Thus:

$$\mathbf{P}(T > \mu n^{\epsilon\gamma}) \geq \frac{1}{4}$$

whence $\tau_{\mathcal{A}} = E_{\mathcal{A}}(T) \geq (1/4)\mu n^{\epsilon\gamma}$. □

Now for the positive result. Let $\mathcal{G}$ denote the following decentralized algorithm:

- At each step X_t, choose among the local neighbors and the shortcut the node u such that $d(u,z)$ is minimized. Let this be X_{t+1}.
- Terminate when z is reached.

This is a known as *greedy routing*. We let $g(u,v) = E_{\mathcal{G}}(T \,|\, Y = u, Z = v)$ be the *greedy distance* from u to v.

Theorem 1.3. *If $\alpha = 1$, then for all vertices u and v, $g(u,v) \leq k_3(\log n)^2$.*

Proof. Like before, we start by bounding the normalizer, Δ_1:

$$\begin{aligned}\sum_{v \neq u} d(u,v)^{-1} &= \sum_{i=1}^{n-1} i^{-1} \\ &\leq 1 + \log(n-1) \leq \kappa \log(n)\end{aligned}$$

for some constant κ. For the proof, we divide the graph into "phases" with respect to a vertices distance from z. We let each phase, $F_j = \{v : 2^j \leq d(v,z) < 2^{j+1}\}$.

Now, assume that $X_t \in F_j$, $\log_2(\log_2(n)) < j \leq \log_2(n)$. We wish to find the probability that we will escape this phase with the next step, ie that $X_{t+1} \notin F_j$. This will occur if the vertex at X_t has shortcut with destination v s.t. $d(v,z) \leq 2^j$. Thus

$$\begin{aligned}\mathbf{P}(X_{t+1} \notin F_j \,|\, X_t \in F_j) &\geq \sum_{d(v,z) \leq 2^j} \frac{1}{\Delta_1 d(X(t), v)} \\ &\geq 2^j \frac{1}{\Delta_1 2^{j+1}} \\ &\geq \frac{1}{2\kappa \log(n)}\end{aligned}$$

Now let T_j is the number of steps spent in phase j. Since we will, in each step, find a shortcut taking us out of the phase with probability at least $1/(2\kappa \log(n))$, and the shortcut at each vertex is selected independently, it holds that, for $\log_2(\log_2(n)) < j \leq \log_2(n)$:

$$E[T_j] \leq 2\kappa \log(n).$$

For $j \leq \log_2(\log_2(n))$ a similar bound holds, possibly after modifying κ, since we can spend at most one step at each vertex. It then follows trivially that

$$E[T \,|\, Y = u, Z = v] \leq \sum_{j=0}^{\log_2 n} T_j \leq \log_2(n) 2\kappa \log(n) = k_3(\log n)^2.$$

□

1.5 Summary of Contributions

This thesis consists of two works, both starting out from Kleinberg's rsults. The first, which is the primary work, discusses the nature and formation of navigable small-world networks. In it, we propose a distributional requirement, conceptually different from Kleinberg's, that also allows greedy routing in $O(\log^2 n)$ time. This requirement relates the probability that a vertex u has a shortcut to another, v, with the probability that queries with destination v visit u. This relationship generalizes naturally to any graph (although the proofs presented do not always do so), and also leads us to propose a stepwise re-wiring algorithm with similar marginal distributions. This algorithm provides an interesting example of how navigable networks may arise naturally.

The second paper tackles the problem of trying to route in Kleinberg type networks if vertices do not start out with global knowledge about their own and others position in the grid. We propose a Markov Chain Monte Carlo algorithm where nodes discover their positions, in a manner than then makes greedy routing possible. The algorithm may have important applications to the development of secure peer-to-peer communication networks, and has therefore been the subject of much popular attention.

Chapter 2

Neighbor Selection[1]

2.1 Introduction

2.1.1 Shortcut Graphs

Starting with the small-world model of Watts and Strogatz, rewired graphs have been the subject of much interest. Such graphs are constructed by taking a fixed graph, and randomly rewiring some portion of the edges. Later models of partially-random graphs have been created by taking a fixed base graph, and adding "long-range" edges between randomly selected vertices (see [29] [31]). The "small-world phenomenon", in this context, is that graphs with a high diameter (such as a simple lattice) attain a very low diameter with the addition of relatively few random edges.

Jon Kleinberg [23] studied such graphs, primarily ones starting from a two dimensional lattice, from an algorithmic perspective. He allowed for $O(N)$ long-range edges, and found that not only would this lead to a small diameter (which was not surprising), but also that if the probability of two nodes having a long-range edge between them had the correct relation to the distance between them in the grid, the *greedy routing* pathlength between vertices was small as well. Greedy routing means, as the name implies, starting from one

[1]This chapter is partly based on joint work with Ian Clarke, who originally proposed the link updating scheme discussed in the later sections in conversation with the current author.

vertex and searching for another by always stepping to the neighbor that is closest to the destination. That the base graph is connected means that a non-overlapping greedy path always exists, so the question regards the utility of the long range contacts in shortening this path. Networks where one can quickly route between two points using only local information at each step, as with greedy routing, are referred to as *navigable*.

For added simplicity, it is advantageous to replace the two dimensional lattice used by Kleinberg with a one dimensional ring of vertices, and move to the directed case where edges follow a single orientation. This means that the lattice distance is the number of steps following the orientation of the ring from one vertex to another - the distance from a vertex to the one "before" it is thus $N - 1$ for a graph of size N. Bariere et al. [5] have performed a thorough investigation of this setting, and calculated the order of the greedy path length for when the probability of a long range contact edge existing between two vertices x and y is $H_N d(x, y)^{-r}$ (d denotes lattice distance, H_N is a normalizing constant). The case $r = 1$ here corresponds to the single critical, navigable case of Kleinberg's model where greedy routing performs in $O(\log^2 n)$ steps, other values of r all lead to greedy path-lengths that are not polynomial in $\log N$.

Initially, we will stay in the one-dimensional directed environment for our work below. Later sections extend some of the results to a wider class of graphs. In general, we will call graphs of the type discussed *shortcut graphs* and use the less clumsy term *shortcut* for the long range contact edges.

2.1.2 Contribution

While Kleinberg's results are important and have been a catalyst for much study, it is not fully understood how the rather arbitrary distribution of shortcuts that they dictate might arise in practice. In this work, we present an alternative distributional requirement that associates the shortcut distribution with the hitting probabilities of queries under greedy routing. We show that distributions that meet this criterion, which we call "balanced distributions" have $O(\log^2 n)$ mean routing times, similarly to the critical case in Klein-

berg's model.

The relationship in this criterion naturally leads to a stepwise re-wiring algorithm for shortcut-graphs. The Markov chain on the set of possible shortcut configurations defined by this algorithm can easily be seen to have a stationary distribution with balanced marginals. While the previous results cannot be directly applied to this case, because the stationary distribution has dependencies between the shortcuts at nearby nodes, we argue through heuristics and simulation that these dependencies in fact work in our favor, and that networks generated by our algorithm can be efficiently navigated.

2.1.3 Previous Work

In [24], Jon Kleinberg himself motivated why the necessary distribution for navigability might arise in nature by means of "group memberships". He showed that in a more generalized setting, structures are navigable if two nodes are connected with a probability that is inversely proportional the size of the smallest group they both populate. That this should be the case is in some sense natural, since the probability of knowing somebody may decrease with the size of the group in which you know them. Similar arguments can be found in [26] and [36].

A paper by Clauset and Moore [11] presents a different re-wiring algorithm for the creation of navigable networks. Rather than associating shortcuts with the destinations of queries that hit a node, they associate then with the end-points of queries that have not found their destination within some threshold number of steps. They show positive results for this algorithm using simulation, but do not present any analytic results. In [15] a re-wiring algorithm for the creation of so called scale-free (or power-law) graphs is presented. This does not deal with clustering nor navigability, and no analytic results regarding the stationary distribution are derived.

The Freenet peer-to-peer data network, presented in [9] and [10], uses a similar method to update the links between peers as the algorithm we propose here. The current work is in part inspired by trying to apply the ideas from the design of Freenet to an environment more conductive to analysis. [39] previously related Freenet to

the discussion of navigable small-world networks, but they worked mostly on proposing modifications to the algorithm that resulted in a more robust network, instead of looking more closely at the properties of Freenet's neighbor sampling.

2.2 Results and Discussion

2.2.1 Distribution and Hitting Probability

We begin by considering some aspects of the hitting probabilities of greedy walks in shortcut graphs. In this section we study only the case where the base graph is a directed cycle, and the shortcuts are additional directed edges. We will index the set of vertices V such that the edges of the base graph are negatively oriented, in the sense that there is an edge from x to $x-1 \mod N$ for all $x = 0 \dots n-1$. The function $d(x, y)$ gives the distance in the base graph from x to y. It is not symmetric, for example $d(x, x-1) = 1$ while $d(x-1, x) = N-1$.

On top of this base graph, we will add one directed shortcut starting at each vertex. We let γ be a configuration of such shortcuts, that is $\gamma : V \to V$. We let Γ be the set of all possible configurations, and we call probability measures on that set *shortcut distributions*.

Given such a shortcut distribution, we define $X_Z^Y(t)$ as a greedy walk in the network from a uniformly chosen starting point $Y = X_z^Y(0)$ with a uniformly chosen destination Z. Below, we will in particular be interested in the hitting probability of greedy walks with specific destinations. We define this formally as:

$$h(x, z) = \mathbf{P}(X_Z^Y(t) = x \text{ for some } t | Z = z) \tag{2.1}$$

Because we are dealing with a transitive base graph and uniform choices of Y and Z, it holds that $h(x, z) = h(d(x, z), 0)$. Thus we will, without loss of generality, discuss only $h(x, 0)$, which we simplify to $h(x)$ below.

Our results concern relating $h(x)$ with the occurrence of shortcuts between nodes. Immediately, however, we can see that $h(x)$ gives us the expected length of a greedy path. Since such a path can hit each

point only once, it follows that if T is the length of a greedy path from a random point to zero, then

$$T = \sum_{x=1}^{N-1} \chi_{\{X_0^Y(t)=x \text{ for some } t\}}$$

whence it follows that:

$$E(T) = \sum_{x=1}^{N-1} h(x). \tag{2.2}$$

We will call the expected greedy walk length $\tau = E[T]$.

We can also prove the following:

Lemma 2.1. *If the shortcut configuration is chosen according to a translation invariant joint distribution, then $h(x)$ is non-increasing in x.*

Proof. Let $I \subset \Gamma \times V$ be event consisting of all configurations and starting points such that a greedy walk for 0 hits the point $x+1$. Now we translate all the coordinates of this set down one coordinate (modulo N), and call the translated set J.

$$h(x+1) = \mathbf{P}(I) = \mathbf{P}(J)$$

by definition and translation invariance. However, every element in J corresponds to a starting point and shortcut configuration for which the greedy walk hits x. To see this, we pick a starting point y and configuration γ, such that $(\gamma, y) \in I$. This means that there is an integer m and a path $x_0, \ldots, x_m$ such that $x_0 = y$, $x_m = x+1$ and either

$$N - 1 \geq \gamma(x_i) > x_i \text{ and } x_{i+1} = x_i - 1$$

or

$$x_i > \gamma(x_i) \geq x + 1 \text{ and } x_{i+1} = \gamma(x_i)$$

for all $i = 0 \ldots m$. The corresponding configuration in J has a similar path $x'_0, \ldots, x'_m$ ($x'_i = x_i - 1$) where $x'_0 = y - 1$, $x'_m = x$ and either:

$$N - 2 \geq \gamma(x'_i) > x'_i \text{ and } x'_{i+1} = x'_i - 1$$

or

$$x'_i > \gamma(x'_i) \geq x \text{ and } x'_{i+1} = \gamma(x_i)'$$

for all $i = 0 \dots m$. This means that starting in $y - 1$ will cause the greedy walk to hit x. (Note that not every configuration and starting point that cause greedy walks to hit x are necessarily in J, since $\gamma(x'_i)$ must be less than $N - 2$ since rather than $N - 1$ in the first line).

It now follows directly that

$$\mathbf{P}(J) \leq h(x).$$

□

We now restrict ourselves to the more manageable case where shortcuts are chosen independently at each point. That is to say that there is some kernel $\ell(x, y)$ such that:

$$\mathbf{P}(\gamma) = \prod_{x \in V} \ell(x, \gamma(x)).$$

We are interested only in kernels which are translation invariant, in other words for which $\ell(x, y) = \ell(d(x, y), 0)$. As with the hitting probability, we will use just $\ell(x)$ to denote $\ell(x, 0)$.

With such a shortcut distribution, we may, for a given z, view $X_z^Y(t)$, as Markov chain on the set of vertices, with some transition kernel $P_z(y, x)$. As above, we will set $z = 0$, and drop the index in the below calculations without loss of generality. The process hits every point except $z = 0$ at most once, and we can let this point be absorbing. The transition kernel P then consists of two mechanisms: either we step to x which is less than y because it is the destination of the shortcut from y, or we step to $y-1$ because y's shortcut overshot 0. (That is, y's shortcut leads to somewhere from which it is further to 0 than y. In other words a point in $\{y+1, \dots, N-1\}$.) It follows that:

$$P(y, x) = \begin{cases} 0 & \text{if } x \geq y \\ \ell(y, x) + \sum_{\xi \geq y+1} \ell(\xi) & \text{if } x = y - 1 \\ \ell(y, x) & \text{if } x \in \{0, \dots, y - 2\} \end{cases}$$

for $y \neq 0$. $P_z(0, x) = \chi_{\{x=0\}}$.

From the transition kernel, we can find a recursive formula for the hitting probability $h(x)$. It follows from the theory of Markov chains that the hitting probability can be written as the solution to a set of equations involving the transition kernel. Using the fact that a greedy walk will never spend more than one time unit in each state non-zero state, this can be written as

$$h(x) = \sum_{\xi} h(\xi)P(\xi, x) + \mathbf{P}(X_0^Y(0) = x).$$

for all x. Using our values for P and noting that the last term is simply $\mathbf{P}(Y = x) = 1/(N-1)$, the above can be written as the recursion formula

$$h(x) = \sum_{\xi=x+1}^{N-1} h(\xi)\ell(\xi - x) + h(x+1) \sum_{\xi=x+2}^{N-1} \ell(\xi) + \frac{1}{N-1} \qquad (2.3)$$

for all $x = 0 \dots N-2$, and with boundary $h(N-1) = 1/(N-1)$. Thus it is possible to find $h(x)$ for a given shortcut distribution ℓ.

2.2.2 Balanced Shortcuts

We now look at a class of shortcut distributions with a certain property. Consider a distribution ℓ such that:

$$\ell(x, z) = \frac{h(x, z)}{\sum_{\xi=1}^{N-1} h(\xi)} = \frac{h(d(x, z))}{\tau} \qquad (2.4)$$

where h is given by (2.1). That is to say that the probability of choosing a shortcut of distance x, is that same as the normalized probability that x is hit when routing for 0. We will call this a *balanced* shortcut distribution.

By plugging (2.4) into (2.3) we can see that for a balanced distribution h (and thus ℓ) must be the solution to an equation of $N-1$ variables.

Distributions that are balanced, it turns out, lead to networks with the same navigability properties as the critical case in Kleinberg's model. Our central result is:

Theorem 2.2. *For every $N = 2^k$ with $k \geq 4$, the shortcut graph with shortcuts selected independently according to a balanced distribution has an expected greedy routing time*

$$\tau \leq 2k^2.$$

The proof method is similar to that of Kleinberg's proof for harmonic links, but the implicit definition of the shortcut distribution requires a somewhat more involved approach.

Proof. Assume that $\tau > 2k^2$. We will show that for k sufficiently large this always leads to a contradiction.

To start with, divide $\{1, \ldots, N-1\}$ into at most k disjoint phases. Each phase is a connected set of points, each successively further from the destination 0, and they are selected so that a greedy walk is expected to spend as many steps in each phase. Thus, the first phase is the interval $F_1 = \{1, \ldots, r_1\}$ where r_1 is the smallest number such that

$$\ell(F_1) = \sum_{\xi \in F_1} \ell(\xi) \geq 1/k$$

The second phase is defined similarly as the interval $\{r_1 + 1, \ldots, r_2\}$: again being the smallest such interval so that $\ell(F_2) \geq 1/k$. Let m be the total number of such intervals which can be formed, and let F_R denote remainder interval $\{r_m + 1, \ldots, N-1\}$, if necessary (let it be the empty set otherwise). By construction $\ell(F_R) < 1/k$ and the total number of phases, including F_R is $\leq k$.

Before proceeding, we need to bound how much ℓ of the different phases can deviate, since this will also tell us how much the expected number of steps in each phase can differ. From (2.4) and the assumed lower bound of τ, it follows that:

$$\ell(x) = \frac{h(x)}{\tau} \leq \frac{1}{2k^2}$$

for all x. This implies that $1/k \leq \ell(F_i) \leq 1/k + 1/(2k^2)$ for all $i \in \{1, \ldots, m\}$, and thus:

$$\ell(F_i) \leq \left(1 + \frac{1}{2k}\right) \ell(F_j) \tag{2.5}$$

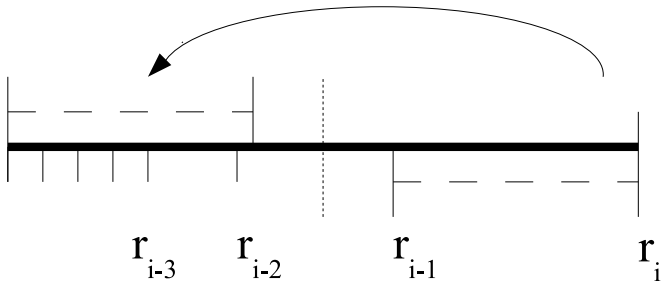

Figure 2.1: Illustration for the proof of Theorem 2.2. If a phase covers less then half of the "remaining ground", then the a shortcut in the equivalent range takes us out of the phase.

for all $i, j \in \{1, \ldots, m\}$. It also gives $m \geq k^2/(k+1) - 1$.

Consider now $F_m = \{r_{m-1} + 1, \ldots, r_m\}$. We know that $r_m < N$. Assume that $r_{m-1} \geq r_m/2$. F_m then covers less than half of the distance from r_m to the target. In particular

$$r_m - r_{m-1} - 1 \leq r_{m-1}$$

so the interval $G = \{0, \ldots, r_m - r_{m-1} - 1\}$ is disjoint with F_m and consists entirely of points which are closer to 0 than those in F_m. Thus, if r_m has a shortcut with destination in this interval, any query which hits r_m will leave F_m in the next step. See Figure 2.1.

We know the probability with which this occurs

$$\ell(r_m, G) = \sum_G \ell(r_m, \xi) = \ell(F_i) \geq 1/k.$$

Lemma 2.1 tells us that the probability of having a shortcut to G cannot decrease for points less than r_m, so for each vertex the query hits within F_m, there is an independent probability of $1/k$ of leaving

F_m in the next step. This means that the expected number of steps the query can take in F_m is at most k.

The expected number of steps in a phase, $h(F_i) = \tau\ell(F_i)$, so by (2.5) it then holds that:

$$h(F_i) \leq (1 + 1/2k)h(F_m) \leq k + 1/2 \tag{2.6}$$

for all $i \in \{1, \ldots, m\}$ and also for F_R. There are at most k phases, so this implies that $\tau \leq k^2 + k/2$, which contradicts our assumption for all $k \geq 2$.

Thus the original assumption implies that $r_{m-1} \leq r_m/2 \leq N/2$. But by an identical argument for F_{m-1}, we can show that $r_{m-2} \leq r_{m-1}/2$. It follows by iteration that

$$r_i \leq \frac{1}{2^{m-i}} N.$$

and specifically:

$$r_1 \leq \frac{1}{2^{m-1}} N \leq 2^{\frac{k+2}{k+1}} \leq 4.$$

This means that F_1 contains at most 4 points, which means that $h(F_1) \leq 4 \leq k$ for $k \geq 4$, and by the argument in (2.6), $\tau \leq k^2 + k/2^2$. This again contradicts the original assumption. The result follows. □

Theorem 2.2 gives us an alternate distributional criterion for attaining $O(\log^2 N)$ expected greedy pathlengths. Since Kleinberg showed that this cannot hold for most distributions, the balanced distributions must be close to the critical, harmonic case.

2.2.3 Other Graphs

We attempt to see how Theorem 2.2 can be generalized to shortcut graphs on more general base graphs than the circle. For the preliminary results to hold, we need to limit ourselves to classes of finite transitive graphs. The most simple examples of such graphs are,

[2]It may seem strange that we are here using that a constant is $O(k)$. In fact, this shows that the bound in the theorem could strengthened somewhat, though it would have the same dominant order in k.

except for the circle itself, toric lattices of higher dimensions with the same circumference in each dimension. While we used a directed ring for simplicity previously, we cannot have a directed base graph in higher dimensions (it is easy to see that not even Kleinberg's can hold if you do), and thus use an undirected base.

In order to use a proof like that of Theorem 2.2 on a class of graphs we need the following property. For a given graph G, we let $B_r(x)$ be a ball of radius r around a vertex x, that is

$$B_r(x) = \{y \in V \text{ s.t. } d_G(x, y) \leq r\}$$

where $d_G(x, y)$ is the geodesic distance in G. Equivalently let $S_r(x) = \partial B_r(x)$, the sphere of radius r around x in the graph.

Definition 2.3. *A class of graphs is called* fair*, if there exist $a \in (0, 1)$ and $c \in (0, 1]$, such that for any graph G in the class:*

$$|S_r(x) \cap B_q(0)| \geq c|S_r(x)|$$

for all $x \in V$, and $ad_G(x, 0) = q \leq r \leq d_G(x, 0)$.

That is to say: if q is a fraction a of the distance from a vertex x to 0, a sphere of radius at least q around x intersects a ball of radius q around 0 on at least a fixed portion of its points.

On top of the base graph, we add a configuration of directed shortcut edges as before. The definition of hitting probabilities and balanced distributions are also the same.

For a fair class of base graphs, with fixed values for a and c, a similar argument to the proof of Theorem 2.2 can be made. Given a base graph, divide the space into approximately $\log_{1/a} N$ "rings" around 0 where we expect to spend as much time in each one. That is let the first ring have the form:

$$F_1 = \bigcup_{r=1}^{r_1} S_r(0)$$

where r_1 is smallest value such that $\ell(F_1) \geq 1/\log_{1/a} N$. And then the other rings as above. Since $h(S_r(0)) \leq 1$ for all r, ℓ of each phase will again be approximately the same.

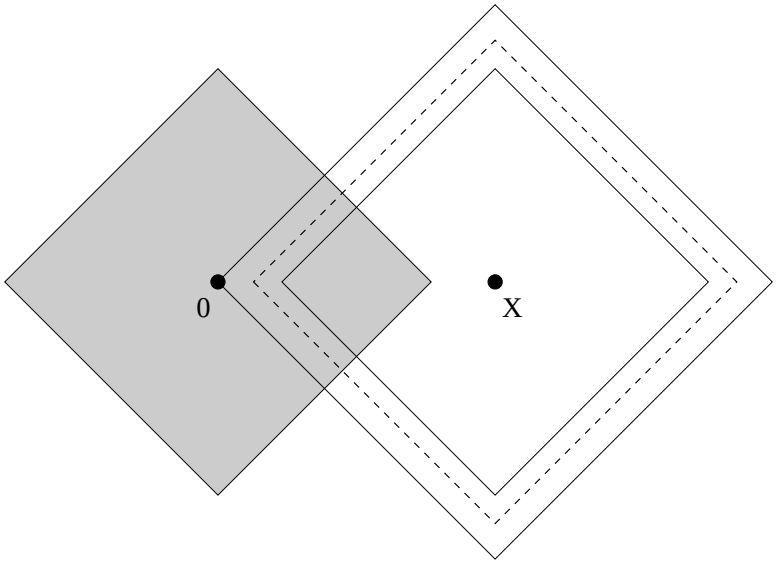

Figure 2.2: A illustration of the "fairness" property of a class of graphs. Every circle (the dotted line) of radius between $q = ad(x, 0)$ and $d(x, 0)$ must have some portion of its vertices within q of 0.

Now, if such a ring has outer radius d and inner radius greater than ad then by "fairness" there is a shortcut leading to a point in a phase closer to 0 with at least probability $c \log_{1/a} N$ in each step. Thus can we spend only a logarithmically bounded time in the ring. If this holds for one ring it holds for all, if all rings have an outer radius that is $1/a$ of their inner radius, the smallest must have radius bounded a constant. A similar contradiction is to the one above is thus derived, and thus a bound of order $(\log n)^2$ is found for all graphs in the class.

It is relatively easy to see that the square grids ($\mathbb{Z}^d \mod N$) are balanced. See for instance Figure 2.2 for the natural intuition. A proof method is sketched, formalizing everything is tedious but not difficult.

Lemma 2.4. *For every k, the class graphs of finite, toric, k-dimensional square grids are fair, with $a = 3/4$ and $c \geq (2k4^{k-1})^{-1}$.*

Proof. (Sketch) Fix such a graph, and let d be its distance function. Let $\delta = d(x, 0)$ and let z be a point vertex halfway between them on a minimal path. Construct $S_{\frac{1}{4}\delta}(z)$. All the points on this sphere lie within $a\delta$ of 0, and at least one side, and thus at least

$$(\frac{1}{4}\delta)^{k-1}$$

vertices, lie on the circle $S_{a\delta}(x)$. By moving this side "towards" 0, we can keep it on $S_r(x)$ for all $a\delta \leq r \leq \delta$ while keeping all of its points within $a\delta$ of 0. Thus at least

$$(\frac{1}{4}\delta)^{k-1}/(2d\delta^{k-1}) = c$$

of the points on any such sphere lie in $B_{a\delta}(0)$. □

Whether it can be shown that other classes of graphs, perhaps all that are generated from subsets of a transitive and amenable infinite graph, have the property we have called fairness is currently an open question to us.

2.3 Re-wiring Algorithm

In this section, we propose an algorithm for the re-wiring of shortcut graphs of the type described above. Running the algorithm modifies, in each step, the destinations of the shortcut edges of vertices in the graph in a random fashion. It is a steady-state algorithm in the sense that it neither creates nor destroys edges: it simply shifts the destinations of the single existing shortcut at each vertex.

In the sense that we propose a generative process which might explain why navigable networks arise, this is similar to the celebrated preferential attachment model for power law networks of Barabási and Albert. However, it is a not a growth model for the network since the number of nodes and edges never changes, and is thus more similar to the model discussed in [15].

The proposed algorithm is as follows:

Algorithm 2.5. *Let (V, E_s) be the directed graph of shortcuts at time s. From each vertex there is exactly one edge. Let $0 < p < 1$. Then (V, E_{s+1}) is defined as follows.*

1. *Choose y_{s+1} and z_{s+1} uniformly from V.*
2. *If $y_{s+1} \neq z_{s+1}$, do a greedy walk from y_s to z_s along the lattice and the shortcuts of E_s. Let $x_0 = y_{s+1}, x_1, x_2, ..., x_t = z_{s+1}$ denote the points of this walk.*
3. *For each $x_0, x_1, ..., x_{t-1}$ independently with probability p replace its current shortcut with one to z_{s+1}.*

After a walk is made, E_{s+1} is the same as E_s, except that the shortcut from each node in walk $s+1$ is with probability p replaced by an edge to the destination. In this way, the destination of each edge is a sample of the destinations of previous walks passing through it. The claim is that updating the shortcuts using this algorithm eventually results in a shortcut graph with greedy pathlengths of $O(\log^2 n)$.

The value of p is a parameter in the algorithm. It serves to disassociate the shortcut from a vertex with that of its neighbors.

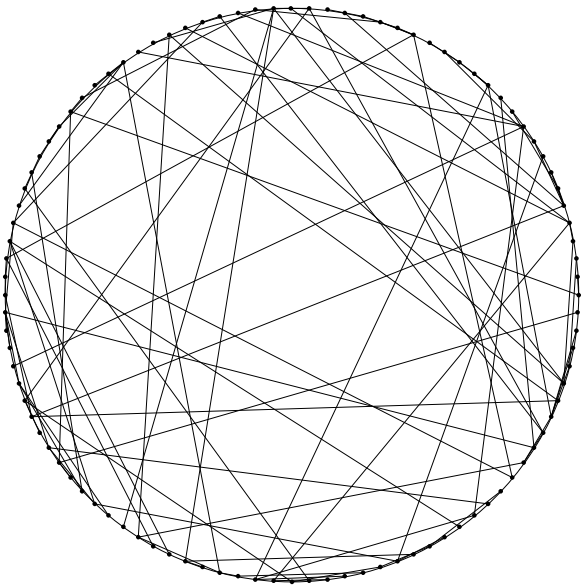

Figure 2.3: A shortcut graph generated by our algorithm ($N = 100$).

For this purpose, the lower the value of $p > 0$ the better, but very small values of p will also lead to slower sampling. It is hard to state an optimal value for p but there are simple heuristic arguments for why p should reasonably be on the order of one over the expected length of the greedy walks.

2.3.1 Computer Simulation

Simulations indicate that the algorithm gives results which scale as desired in the number of greedy steps, and that the distribution approximates $H_N/d(x, y)$.

The results in the directed one-dimensional case can be seen in Figure 2.4. To get these results, the network is started with no shortcuts, and then the algorithm is run $10N$ times to initialize the references. The value of $p = .10$ is used. The greedy distance is then measured as the average of 100,000 walks, each updating the graph according to the algorithm. The effect of running the algorithm,

rather than freezing one configuration, seems to be lower the variance of the observed value.

The square root of the mean greedy distance increases linearly as the network size increases exponentially, just as we would expect. In fact, as can be seen, our algorithm leads to better simulation results than choosing from Kleinberg's distribution. Doubling the network size is found to increase the square route of the greedy distance by circa 0.41 when links are selected using our algorithm, compared to an increase of about 0.51 when Kleinberg's model is used. (In fact, in with Kleinberg's model we can use (2.3) to calculate numerically exact values for τ, allowing us to confirm this figure.)

In Figure 2.4 the marginal distribution of shortcut lengths is plotted. It is roughly harmonic in shape, except that it creates less links of length close to the size of the network. This may be part of the reason why it is able to outperform Kleinberg's model: while Kleinberg's model is asymptotically correct, this algorithm takes into account finite size effects. (This reasoning is similar to that of the authors of [11]. Like them, we have no strong analytic arguments for why this should be the case, which makes it a tenuous argument at best.)

The algorithm has also been simulated to good effect using base graphs of higher dimensions. Figure 2.5 shows the mean greedy distance for two dimensional grids of increasing size. Here also, the algorithm creates configurations that seem to display square logarithmic growth, and which perform considerably better than explicit selection according to Kleinberg's model.

2.3.2 Markov Chain View

Each application of Algorithm 2.5 defines the transition of a Markov chain on the set of shortcut configurations, Γ. The Markov chain in question is defined on a finite (if large) state space. If it is irreducible and aperiodic, it thus converges a unique stationary distribution.

Theorem 2.6. *The Markov chain $(E_s)_{s\geq 0}$ is irreducible and aperiodic.*

Proof. Aperiodic: There is a positive probability that $y_s = z_s$ in which case nothing happens at step s.

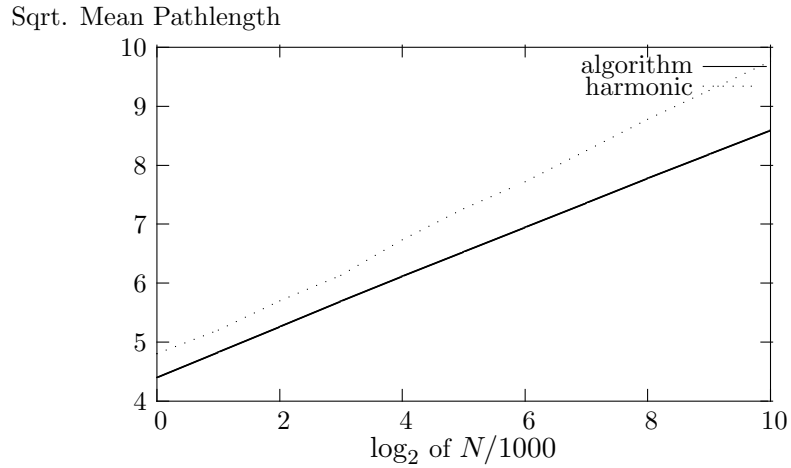

Figure 2.4: Data from the tables in Section 2.3.1 on the expected greedy walk length using our selection algorithm, compared to selection according the harmonic distribution.

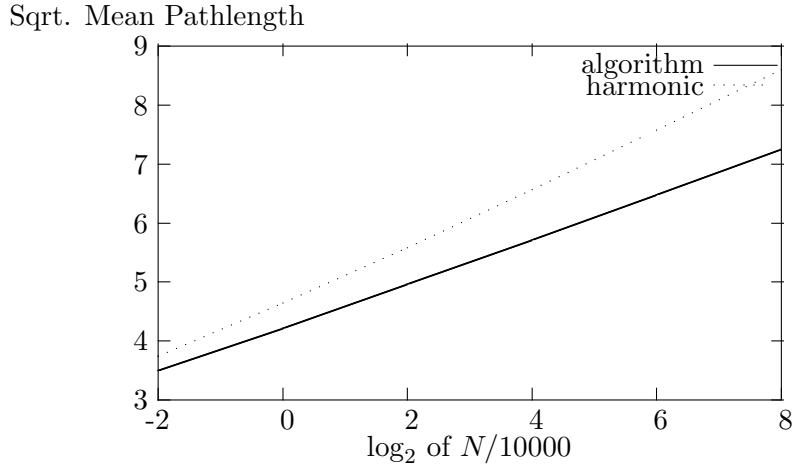

Figure 2.5: The expected greedy walk time of the selection algorithm, compared to selection according to harmonic distances, in a two dimensional base grid.

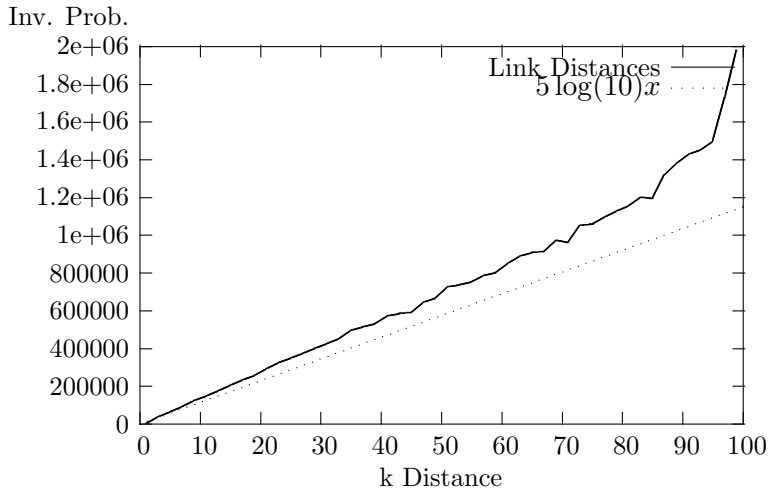

Figure 2.6: The inverse of distribution of shortcut distances, with $N = 100000$, $p = 0.10$. The straight line is the inverse of the harmonic distribution.

Irreducible: We need to show that there is a positive probability of going from any shortcut configuration to any other in some finite number of steps. This follows directly if there is a positive probability that we can "re-point" the shortcut starting at a vertex x to point at a given target y without changing the rest of the graph. But the probability of this happening in a single iteration is:

$$\geq \frac{1}{N}\frac{1}{N}p(1-p)^{N-2} > 0.$$

□

Thus there does exist a unique stationary shortcut distribution, which assigns some positive probability to every configuration. The goal is to motivate that this distribution leads to short greedy walks.

We can look at the marginal distribution of the shortcut distances at every point. The shortcut from a vertex x at any time is simply a sample of the destination of the previous walks that x has seen. Under the stationary distribution this should not change with time,

so

$$\ell(x,z) = \mathbf{P}(Z = z | X_Z^Y(t) = x \text{ for some } t).$$

Using Bayes' theorem, we can related this to the hitting probability.

$$\begin{aligned}\ell(x,z) &= \mathbf{P}(Z = z | X_Z^Y(t) = x \text{ for some } t) \\ &= \frac{\mathbf{P}(X_Z^Y(t) = x \text{ for some } t | Z = z)\mathbf{P}(Z = z)}{\sum_{\xi \neq x} \mathbf{P}(X_Z^Y(t) = x \text{ for some } t | Z = \xi)\mathbf{P}(Z = \xi)}\end{aligned}$$

The first multiple in the numerator is the hitting probability $h(x,z)$. It then follows from the uniform distribution of Z that:

$$\ell(x,z) = \frac{h(x,z)}{\sum_{\xi \neq x} h(x,\xi)} = \frac{h(x,z)}{\sum_{\xi=1}^{N-1} h(\xi)}$$

This shows that the marginal shortcut distribution at each point under the stationary distribution is balanced, and it is tempting to apply Theorem 2.2. However, that theorem assumed that the shortcuts had been chosen independently at each vertex, which is not the case here.

There are two sources of dependencies between the shortcuts of neighboring vertices. Firstly, there is a chance that they sampled the destination of the same walk. When p is large, this dependency is substantial, and we see a highly detrimental effect even in the simulations. By using a small p, however, this dependence is muted. Another, more subtle dependence, has to do with the way the shortcuts of vertices around a vertex x may affect the destinations of the walks it sees. If $x+1$ has a shortcut to $x-10$, that will make it less likely for x see walks for places "beyond" $x-10$ since many such walks will have followed the shortcut at $x+1$, and thus skipping over x.

The first dependence, that of sampling from the same walk, can be handled by modifying the algorithm to make sure we do not sample more than once for each walk. Take $p \leq 1/N$ and once a walk is completed, we choose to update exactly one of its links with probability pw where w is the length of the walk. Which link to update is then chosen uniformly from the walk. This way, the probability a vertex updates its shortcut when hit by a walk is still always p, but

we never sample two shortcuts from the same walk. The modified algorithm is less natural, but clearly a good approximation of the original for small p values. Although it is more complicated, it is probably not harder to analyze, since it allows for the simplifying assumption that each edge is chosen from a different greedy walk.

The other dependencies are more complicated, and there is no easy way to modify the algorithm to remove them. However, it is worth noting that it is hard to see why these dependencies (unlike the first type) would be destructive for greedy routes. In fact, it makes sense that if x in our example gets few walks destined beyond $x - 10$ because of the shortcut present at $x + 1$, then it should also choose a shortcut to beyond $x - 10$ with a smaller probability.

In the proof of Theorem 2.2 we use independence only to show that if the probability of having a shortcut out of a phase at the very furthest point is ρ, then the expected steps in the phase is bounded by $1/\rho$. There is little reason to believe this wouldn't hold under the algorithm, since if the link from the furthest point doesn't take us out the phase, it either goes to a point within the phase, or overshoots the destination. If it goes to a point within the phase, then we follow it, and the presence of that shortcut should not interfere with the shortcut from the destination. If it, on the other hand, overshoots, then by the above argument it should make it more likely that the following ones don't overshoot, giving a us a better than independent probability of leaving the phase.

Formalizing the requirements on the dependence, and proving that our stationary distribution indeed agrees with them, is the main open problem left to resolve about this work.

2.4 Conclusion

The study of navigable networks is still in its infancy, but many interesting results have already been found, and the practical relevance to such fields as computer networks is beyond doubt. In this paper we have presented a different way of looking at the dynamics that cause networks to be navigable, and we have presented an algorithm which may explain how navigable networks arise naturally. The algorithm's

simplicity also means that it can be useful in practice for generating networks that can easily be searched, and important property for many structures on the Internet.

While many questions about these networks in general, and our algorithm in particular, remain unanswered, the prospects of going further with this work seem good. We are hopeful that these ideas will be fruitful, leading to further analysis of searching and routing in networks of all kinds.

Chapter 3

Distributed Routing[1]

3.1 Introduction

The modern view of the so called "small-world phenomenon" can be dated back to the famous experiments by Stanley Milgram in the 1960s [28]. Milgram experimented with people's ability to find routes to a destination within the social network of the American population. He concluded that people were remarkably efficient at finding such routes, even towards a destination on the other side of the country. More recent studies using the Internet have come to the same conclusion, see [13].

Models to explain why graphs develop a small diameter ([37], [8], [35]), have been around for some times. Generally, these models specify the mixing of a structured base graph, such a as grid, and random "shortcuts" edges between nodes. However, it was not until Jon Kleinberg's work in 2000 [23] that a mathematical model was developed for how efficient routing can take place in such networks. Kleinberg showed that the possibility of efficient routing depends on a balance between the proportion of shortcut edges of different lengths with respect to coordinates in the base grid. Under a specific distribution, where the frequency of edges of different

[1]This chapter is due to be presented, as "Distributed Routing in a Small World" at the SIAM ALENEX Workshop on experimental algorithms in January 2006. I would like to thank the reviewers for their input.

lengths decreases inverse proportionally to the length, simple greedy routing (always walking towards the destination) can find routes in $O(\log^2(n))$ steps on average, where n is the size of the graph.

3.1.1 Motivation

Kleinberg's result is sharp in the sense that graphs where edges are chosen from a different distribution are shown not to allow for efficient searching. However, the small-world experiments seem to show that greedy-like routing is efficient in the world's social network. This indicates that some element of Kleinberg's model is present in the real world. In [24] and [36] this is motivated by reason of people's group memberships[2]. Several dynamic processes by which networks can evolve to achieve a similar edge distribution have also been proposed recently, for example, in [11], as well as in forthcoming work by this author [32].

However, in Kleinberg's search algorithm, the individual nodes are assumed to be aware of their own coordinates as well as those of their neighbors and the destination node. In the case of real world data, it may be difficult to identify what these coordinates are. In fact the participant nodes may be unaware of anything but their immediate neighborhood and thus oblivious of the global structure of the graph, and, importantly for this work, of geographic (or other) coordinates. For example, in peer-to-peer overlay networks on the Internet, one may wish to automatically find routes without relying on information about the local user, let alone his neighbors or the routes target. In such a situation, how can we search for short paths from one node to another?

3.1.2 Contribution

With this in mind, this paper attempts to return to Milgram's original problem of finding paths between people in social networks. Starting from an unmarked shortcut graph and no other information on the coordinates, we attempt to fit it against Kleinberg's model

[2]Roughly: When a group is twice as large, people in it are half as likely to know each other.

so as to make efficient searches possible. Taking as hypothesis that the graph was generated by applying Kleinberg's distribution model to a base graph with co-ordinate information, we attempt to recover the *embedding*. We approach this as a statistical estimation problem, with the configuration of positions in the grid assigned to each node as a (multi–dimensional) unknown parameter. With a good estimate for this embedding, it is possible to make greedy routing work without knowing the original positions of the nodes when the graph was generated. We employ a Markov Chain Monte-Carlo (MCMC) technique for fitting the positions.

We summarize our contributions as follows:

1. We give an MCMC algorithm to generate an embedding of a given graph into a one or two dimensional (toric) grid which is tuned to the distributions of Kleinberg's model.

2. This method is tested using artificially generated and controlled data: graphs generated according to the ideal model in one and two dimensions. The method is demonstrated to work quite well.

3. It is then applied to real social network data, taken from the "web of trust" of the users of an email cryptography program.

4. Finally, it is observed that the method used can be fully distributed, working only with local knowledge at each vertex. This suggests an application to routing in decentralized networks of peers that only connect directly to their own trusted friends in the network. Such networks, known as Friend-to-Friend networks of Darknets, have so far been limited to communication only in small cliques, and may become much more useful if global routing is made possible.

5. Our algorithm can thus be viewed also as a general purpose routing algorithm on arbitrary networks. It is tailored to "small world" networks, but appears to also work quite well for a more general class of graphs.

3.1.3 Previous Work

Different methods of searching social networks and similar graphs have been discussed in previous work. In [3] a method is proposed for searching so called "power-law networks", either by a random walk or by targeting searches at nodes with high degree. Because such graphs have a highly skewed degree distribution, where a small set of nodes are connected to almost everyone, the methods are found to work well. The first author of that paper and a co-author recently investigated the problem of searching social networks in [2]. There they found that power-law methods did not work well, and instead attempted to use Kleinberg's model by trying to identify people's positions in some base graph based on their characteristics (where they live, work, etc). This was found to work well on a network with a canonical, highly structured base graph (employees of Hewlett Packard) but less well on the social network of students at Stanford University. Similarly Liben-Nowell et. al. [26] performed greedy searches using the town names as locations in the network of writers on the website "LiveJournal". They claim positive results, but consider searches successful when the same town as the desired target is reached: a considerably easier task than routing all the way.

In [38] the authors attempt to find methods to search a network of references between scientific authors. They mention Kleinberg's model, but state:

> "The topology of referral networks is similar to a two-dimensional lattice, but in our settings there is no global information about the position of the target, and hence it is not possible to determine whether a move is toward or away from the target".

It is the necessity of having such information that we attempt to overcome here.

3.2 Kleinberg's Model

Kleinberg's small-world model, like that of Watts and Strogatz [37] which preceded it, starts with a base graph of local connections,

onto which a random graph of shortcut edges (long range contacts) is added. In its most basic form, one starts with a k-dimensional square lattice as the base network, and then adds q directed random edges at each node, selected so that each such shortcut edge from x points to y with probability:

$$\ell(x, y) = \frac{1}{\mathrm{d}(x, y)^k H_k(n)}$$

where d denotes lattice distance in the base graph, n the size of the network, and H_k is a normalizing constant.

Kleinberg showed that in this case so-called *greedy routing* finds a path from any point to any other in, on average, $O(\log^2(n))$ steps. Greedy routing means always picking the neighbor (either through a shortcut or the base graph) which is closest to the destination, in terms of the lattice distance d, as the next step. Since routing within the base graph is permitted, the path strictly approaches the destination, and the same point cannot be visited twice.

In order to make the model more applicable to the real world, it is desirable to use the base graph only as a distance function between nodes, and thus only use the shortcut edges when routing. The necessity of a strictly approaching path existing then disappears, and we are left with the possibility of coming to a dead-end node which has no neighbor closer to the destination than itself. Kleinberg himself dealt with this issue in [24], working on non-geographical models, and there used q (node degree) equal to $\kappa \log^2(n)$ for a constant κ. In this case it is rather easy to see that κ can be chosen so as to make the probability that any node in the network is dead-end for a given query is arbitrarily small for all sizes n.

Actually, it suffices to keep the probability that a dead-end is encountered in any given route small. By approximate calculations one can see that this should hold if $q = \Theta(\log(n) \log\log(n))$[3]. In practice we find that scaling the number of links with $\log(n)$ preserves the number of paths that do not encounter a dead end for all Kleinberg model graphs we have simulated.

[3]Roughly: The probability that a link will not be dead-end to a query decreases with $(\log n)^{-1}$. With $c \log(n) \log\log(n)$ links per node, the probability

3.3 The Problem

The problem we are faced with here is this: given a network, presumed to be generated as the shortcuts in Kleinberg's model (in some number of dimensions), but without any information on the position of the nodes, can we find a good way to embed the network into a base grid so as to make the routing between them possible? This may be viewed as a parametric statistical estimation problem. The embedding is thus seen as the model's parameter, and the data set is a single realization of the model.

Seen from another perspective, we are attempting to find an algorithmic approach to answering the fundamental question of greedy routing: which of my neighbors is closest to the destination? In Kleinberg's model this is given, since each node has a prescribed position, but where graphs of this type occur in real life, that is not necessarily the case. The appeal of the approach described below is that we can attempt to answer the question using no data other than the graph of long connections itself, meaning that we use the clustering of the graph to answer the question of who belongs near whom.

Our approach is as follows: we assign positions to the nodes according to the a-posteriori distribution of the positions, given that the edges present had been assigned according to Kleinberg's model. Since long edges occur with a small probability in the model, this will tend to favor positions so that there are few long edges, and many short ones.

3.4 Statement

Let V be a set of nodes. Let ϕ be a function from V onto G, a finite (and possibly toric) square lattice in k dimensions[4]. ϕ is the configuration of positions assigned ! to the nodes in a base graph G. Let d denote graph distance in G. Thus for $x, y \in V$, $\mathrm{d}(\phi(x), \phi(y))$

that a given node is a dead-end is thus bounded by $(\log n)^{\theta}$. θ can be made large by choosing a large c, thus making the probability of encountering a node in the $O(\log n)^2$ nodes encountered in a walk small.

denotes the distance between respective positions in the lattice.

Let E denote a set of edges between points in V, and let them be numbered $1, \ldots, m$. If we assume that the edges are chosen according to the Kleinberg's model, with one end fixed to a particular node and the other chosen randomly, then the probability of a particular E depends on the distance its edges cover with respect to ϕ and G. In particular, if we let x_j and y_j denote the start and end point, respectively, of edge j, then:

$$\Pr(E|\phi) = \prod_{i=1}^{m} \frac{1}{\mathrm{d}(\phi(x_i), \phi(y_i))^k H_G} \tag{3.1}$$

where H_G is a normalizing constant.

When seen as a function of ϕ, (3.1) is the likelihood function of a certain configuration having been used to generate the graph. The most straightforward manner in which to estimate ϕ from a given realization E is to choose the maximum likelihood estimate, that is the configuration $\hat{\phi}$ which maximizes (3.1). Clearly, this is the same as configuration which minimizes the product (or, equivalently, log sum) of the edge distances. Explicitly finding $\hat{\phi}$ is clearly a difficult problem: in one dimension it has been proven to be NP-complete [17], and there is little reason to believe that higher dimensions will be easier. There may be hope in turning to stochastic optimization techniques.

Another option, which we have chosen to explore here, is to use a Bayesian approach. If we see ϕ as a random quantity chosen with some probability distribution from the set of all possible such configurations (in other words, as a parameter in the Bayesian tradition), we can write:

$$\Pr(\phi|E) = \frac{\Pr(E|\phi)\Pr(\phi)}{\Pr(E)} \tag{3.2}$$

which is the a-posteriori distribution of the node positions, having observed a particular set of edges E. Instead of estimating the maximum likelihood configuration, we will try to assign configurations according to this distribution.

[4] In our experiments below, we focus mostly on the one dimensional case, with some two dimensional results provided for comparisson purposes.

3.4.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a remarkable algorithm used in the field of Markov Chain Monte-Carlo. It allows one, given a certain distribution π on a set S, to construct a Markov chain on S with π as its stationary distribution. While simulating a known distribution might not seem extraordinary, Metropolis-Hastings has many properties that make it useful in broad range of applications.

The algorithm starts with a selection kernel $\alpha : S \times S \mapsto [0, 1]$. This assigns, for every state s, a distribution $\alpha(s, r)$ of states which may be selected next. The next state, r, is selected according to this distribution, and then accepted with a probability $\beta(s, r)$ given by a certain formula of α and π. If the state is accepted, it becomes the next value of the chain, otherwise the chain stays in s for another time-step. If r is the proposed state, then the formula is given by:

$$\beta(s, r) = \min\left(1, \frac{\pi(r)\alpha(r, s)}{\pi(s)\alpha(s, r)}\right).$$

The Markov chain thus defined, with transition Matrix $P(s, r) = \alpha(s, r)\beta(s, r)$ if $s \neq r$ (and the appropriate row normalizing value if $s = r$), is irreducible if α is, and can quite easily be shown to have π as its stationary distribution, see [19], [21]. The mixing properties of the Markov chain depend on α, but beyond that the selection kernel can be chosen as need be.

3.4.2 MCMC on the Positions

Metropolis-Hastings can be applied to our present problem, with the aim of constructing a chain on the set of position functions, $S = G^V$, that has (3.2) as its stationary distribution [5]. Let α be a selection kernel on S, and ϕ_2 be chosen by α from ϕ_1. It follows that, if we let $\alpha(\phi_1, \phi_2) = \alpha(\phi_2, \phi_1)$, and assume a uniform a-priori distribution, then:

[5] Another way of looking at this is as an example of *Simulated Annealing*, which uses the Metropolis-Hastings method to try to minimize an energy function. In this case, the energy function is just the log sum of the edge distances, and the β coefficient is 1.

$$\begin{aligned}\beta(\phi_1, \phi_2) &= \min\left(1, \frac{\Pr(E|\phi_2)}{\Pr(E|\phi_1)}\right) \\ &= \min\left(1, \prod_{i=1}^{m} \frac{\mathrm{d}(\phi_1(x_i), \phi_1(y_i))^k}{\mathrm{d}(\phi_2(x_i), \phi_2(y_i))^k}\right)\end{aligned}$$

Let ϕ_2 be an x, y-switch of ϕ_1 if $\phi_1(x) = \phi_2(y)$, $\phi_1(y) = \phi_2(x)$, and $\phi_1(z) = \phi_2(z)$ for all $z \neq x, y$. In such cases, the above simplifies by cancellation to:

$$\beta(\phi_1, \phi_2) = \min\left(1, \prod_{i \in \mathrm{E}(x \vee y)} \frac{\mathrm{d}(\phi_1(x_i), \phi_1(y_i))^k}{\mathrm{d}(\phi_2(x_i), \phi_2(y_i))^k}\right) \tag{3.3}$$

where $\mathrm{E}(x \vee y)$ denotes the edges connected to x or y. This function depends only on edge information that is local to x and y.

We are now free to choose a symmetric selection kernel according to our wishes. The most direct choice is to choose x and y randomly and then to select ϕ_2 as the x, y-switch of ϕ_1. This is equivalent to the kernel:

$$\alpha(\phi_1, \phi_2) = \begin{cases} 1/(n + \binom{n}{2}) & \text{if } x, y\text{-switch} \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

The Markov chain on S with transition matrix

$$P(\phi_1, \phi_2) = \alpha(\phi_1, \phi_2)\beta(\phi_1, \phi_2)$$

with α and β given by (3.4) and (3.3) respectively, is thus the Metropolis-Hastings chain with (3.2) as its stationary distribution. Starting from any position function, it eventually converges to the sought a-posteriori distribution.

A problem with the uniform selection kernel is that we are attempting to find a completely distributed solution to our problem, but there is no distributed way of picking two nodes uniformly at random. In practice, we instead start a short random walk at x, and use as y the node where the walk terminates. This requires no central element. It is difficult to specify the kernel of selection technique explicitely, but we find it more or less equivalent to the one above. See Section 3.8 below.

3.5 Experiments

In order to test the viability of the Markov Chain Monte-Carlo method, we test the chain on several types of simulated data. Working with the one-dimensional case, where the base graph is a circle, we simulate networks of different sizes according to Kleinberg's model, by creating the shortcuts through random matching of nodes, and with the probability of shortcuts occurring inverse squarely proportional to their length. We then study the resulting configuration in several ways, depending on whether the base graph is recreated after the experiment, and whether, in case it is not, we stop when reaching a dead-end node of the type described above.

We also study the algorithm in two dimensions, by simulating data on a grid according to Kleinberg's model, and using the appropriate Markov chain for this case. Finally, we study some real life data sets of social networks, to try to determine if the method can be applied to find routes between real people.

The simulator used was implemented in C on Linux and Unix based systems. Source code, as well as the data files and the plots for all the experiments, can be found at:

`http://www.math.chalmers.se/~ossa/swroute/`

3.6 Experimental Methodology

3.6.1 One-Dimensional Case

We generated different graphs of the size $n = 1000 * 2^r$, for r between 0 and 7. The base graph is taken to be a ring of n points. Each node is then given $3 \log_2 n$ random edges to other nodes. Since all edges are undirected, the actual mean degree is $6 \log_2 n$, with some variation above the base value. This somewhat arbitrary degree is chosen because it keeps the probability that a route never hits a dead end low when the edges are chosen according to Kleinberg's model. Edges are sent randomly clockwise or counterclockwise, and have length between 1 and $n/2$, distributed according to three different models.

1. Kleinberg's model, where the probability that the edge has length d is proportional to $1/d$.

2. A model with edges selected uniformly at random between nodes.

3. A model where the probability of an edge having length d is proportional to $1/d^2$.

Both the latter cases are non-optimal: the uniform case represents "too little clustering", while the inverse square case represents "too much". In Kleinberg's result, the two types of graphs are shown not to have log-polynomial search times in different ways: too much clustering means not enough long edges to quickly advance to our destination, too little means not enough edges that take even closer when we are near it.

Performance on the graphs can be measured in three different ways as well. In all cases, we choose two nodes uniformly, and attempt to find a greedy route between them by always selecting the neighbor closest (in terms of the circular distance) to the destination. The difference is when we encounter a dead end – that is to say a node that has no neighbor closer to the destination then itself. In this case we have the following choices on how to proceed:

1. We can terminate the query, and label it as unsuccessful.

2. We can continue the query, selecting the best node even if it is further from the destination. In this case it becomes important that we avoid loops, so we never revisit a node.

3. We can use a "local connection" to skip to a neighbor in the base from the current node, in the direction of the destination.

For the second case to be practical, it is necessary that we limit the number of steps a query can take. We have placed this limit as $(\log_2 n)^2$, at which point we terminate and mark the query unsuccessful. This value is of course highly arbitrary (except in order), and always represents a tradeoff between success rate and the mean steps taken by successful queries. This makes such results rather difficult to analyze, but it is included for being the most realistic option, in

the sense that if one was using this to try to search in a real social network, the third case is unlikely to be an option, and giving up, as in the first case, is unnecessary.

We look at each result for the graph with the positions as they were when it was generated, after shuffling the positions randomly, and finally with positions generated by a running the Markov Chain for $6000n$ iterations. It would, of course, be ideal to be able to base such a number off a theoretical bound on the mixing time, but we do not have any such results at this time. The number has been chosen by experimentation, but also for practical purposes: for large n the numerical complexity makes it difficult to simulate larger orders of iterations in practical time-scales.

Due to computational limitations, the data presented is based off only one simulation at every size of the graph. However, at least for graphs of limited size, the variance in the important qualities has been seen to be small, so we feel that the results are still indicative of larger trends. The relatively regular behavior of the data presented below strengthens this assessment.

After shuffling and when we continue at dead ends, the situation is equivalent to a random walk, since the greedy routing gains from the node positions. Searching by random walk has actually been recommended in several papers ([3], [18]), so this gives the possibility of comparing our results to that.

3.6.2 Two Dimensional Case

We also simulate Kleinberg's model in two dimensions, generating different graphs of the size $n = 1024 * 4^r$, for r between 0 and 3. A toric grid as the base graph (that is to say, each line is closed into a loop). Shortcuts were chosen with the vertex degrees as above, and with ideal distribution where the probability that two nodes are connected decreasing inverse squared with distance (the probability of an edge having length d is still proportional to $1/d$, but as d increases there are more choices of nodes at that distance). We do this to compare the algorithm in this setting to that in the one dimensional case.

We also try, for graphs with long range connections generated

against a two dimensional base graph, to use the algorithm in one dimension, and vice versa. This is to ask how crucial the dimension of the base grid is to Kleinberg's model: whether the essential characteristics needed for routing carry over between dimensions. Any conclusion on the subject, of course, is subject to the question of the performance of the algorithm.

3.6.3 Real World Data

Finally, we test the method on a real graph of social data. The graph is the "web of trust" of the email cryptography tool Pretty Good Privacy (PGP) [1]. In order to verify that the person who you are encrypting a message for really is the intended recipient, and that the sender really is who he claims to be, PGP has a system where users cryptographically sign each others keys, thereby vouching for the key's authenticity. The graph in question is thus a sample of people that know each other "in real life" (that is outside the Internet), since the veracity of a key can only be measured through face to face contact.

We do not look at the complete web of trust, which contained about 23,000 users, but only at smaller subsets. The reason for this is two-fold. Firstly, the whole network is not a connected component. Secondly a lot of the nodes in the graph are in fact leaves, or have only one or two vertices. Under such conditions, the algorithm (or any greedy routing for that matter) cannot be expected to work.

These were created by starting a single user as the new graph's only vertex, and recursively growing the graph in the following manner. If G_n is the new graph at step n:

1. Let ∂G_n be the vertices with at least one edge into G_n, but who are not in G_n themselves.

2. Select a node x randomly from those members of ∂G_n who have the greatest number of edges into G_n.

3. Let G_{n+1} be the graph induced by the vertices of G_n and x.

4. Repeat until G_{n+1} is of the desired size.

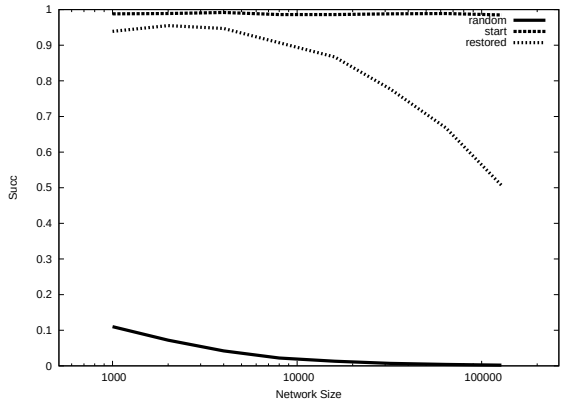

Figure 3.1: The success-rate of queries when terminating at dead-end nodes, on a graph generated by the ideal model.

This procedure is motivated by allowing us to get a connected, dense, "local" subgraph to study. It is closest we can come to the case where, having access to the base graph, one uses a only the nodes in a particular section of it and the shortcuts between them.

Daily copies of the web of trust graph are available at the following URL:

`http://www.lysator.liu.se/~jc/wotsap/`

3.7 Experimental Results and Analysis

3.7.1 One Dimensional Case

Experimental results in the one dimensional case were good in most, but not all, cases. Some of the simulated results can be seen in 3.1 through 3.8. Lines marked as "start" show the values with the graphs as they were generated, "random" show the values when the positions have been reassigned randomly (this was not done for the random matchings case, as there is no difference from the start), and "restored" show the values after our algorithm has been used to optimize the positions.

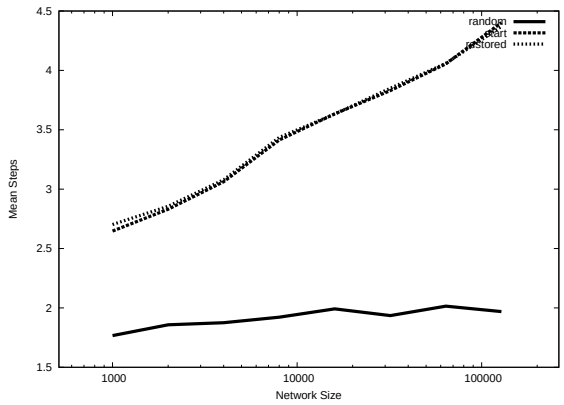

Figure 3.2: Mean number of steps of successful queries when terminating at dead-end nodes, on a graph generated by the ideal model.

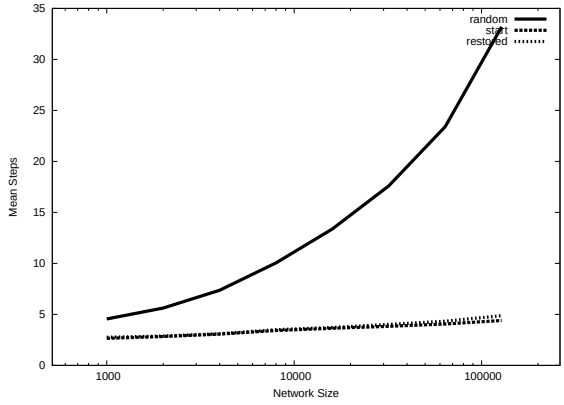

Figure 3.3: Mean number of steps of successful queries when allowed to use local connections, on a graph generated by the ideal model.

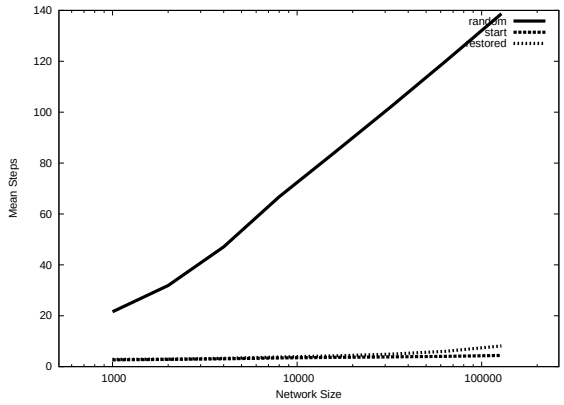

Figure 3.4: Mean number of steps of successful queries when terminating after $(\log_2(n))^2$ steps, on a graph generated by the ideal model.

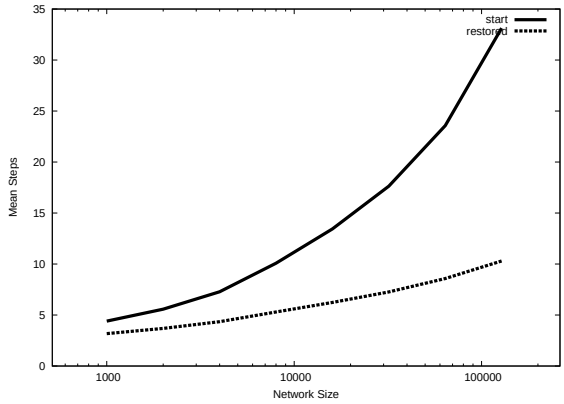

Figure 3.5: Mean number of steps of successful queries when allowed to use local connections, on a graph generated by random matchings.

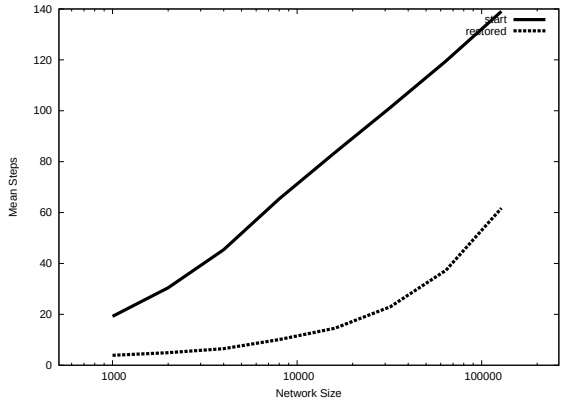

Figure 3.6: Mean number of steps of successful queries when terminating after $(\log_2(n))^2$ steps, on a graph generated by random matchings.

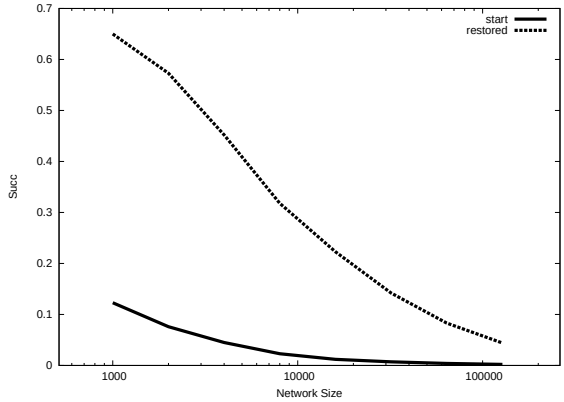

Figure 3.7: The success-rate of queries when terminating at dead-end nodes, on a graph generated by random matchings.

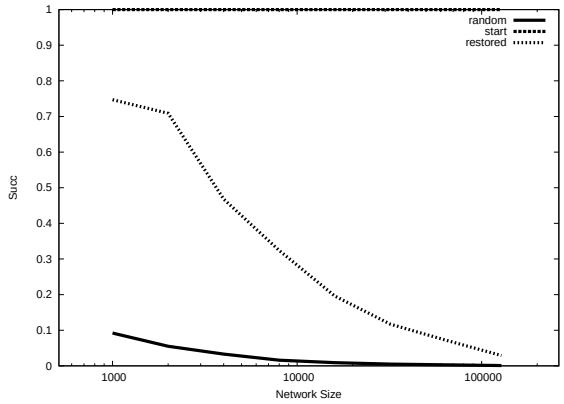

Figure 3.8: The success-rate of queries when terminating at dead-end nodes, on a graph generated with connection probabilities inverse square proportional to the length.

In the ideal graph model, when the original graph is known to allow log polynomial routing, we can see that the algorithm works well in restoring the query lengths. In particular, Figure 3.3, where queries have been able to use the base graph, shows nearly identical performance before and after restoration.

In the cases where queries cannot use the local connections, we see that proportion of queries that are successful is a much harder property to restore than the number of steps taken. Figure 3.1 shows this: for large graphs the number of queries that never encounter a dead-end falls dramatically. A plausible cause for this is that it is easy for the algorithm to place the nodes in the approximately right place, which is sufficient for the edges to have approximately the necessary distribution, but a good success rate depends on nodes being exactly by those neighbors to which they have a lot edges.

Along with the ideal data, two non-ideal cases were examined. In the first case, where the long range connections were added randomly, the algorithm performs surprisingly well. At least with regard to the number of steps, we can see a considerable improvement at all sizes tested. See in particular Figures 3.6 and 3.5. However, it is

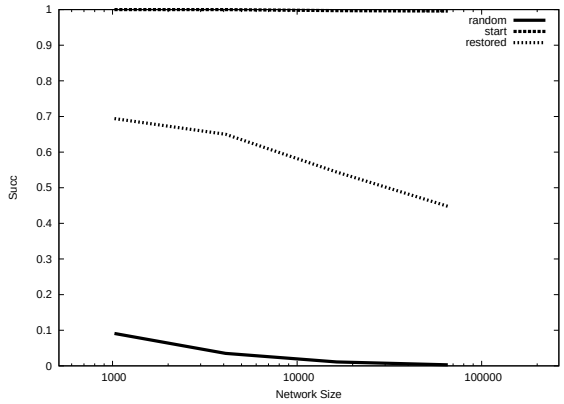

Figure 3.9: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Success rate when failing at dead-end nodes.

impossible for the success rate to be sustained for large networks when the base graph is not used - in this case there simply is no clustering in the graph - and as expected the number of successful queries does fall as n grows (Figure 3.7.

The other non-ideal case, that of too much clustering, was the one that faired the worst. Even though this leads to lots of short connections, which one would believe could keep the success rate up, this was not found to be the case. Both the success rate and the mean number of steps of the successful queries were not found to be significantly improved by the algorithm in this case. The results in Figure 3.8 if particularly depressing in this regard. It should be noted that it has been shown [27] that graphs generated in this way are not small-world graphs - their diameter is polynomial in their size, so there is no reason to believe that they can work well for this type of application.

3.7.2 Two Dimensional Case

The algorithm was also simulated with a pure two dimensional model. In general, the algorithm does not perform as well as in the one di-

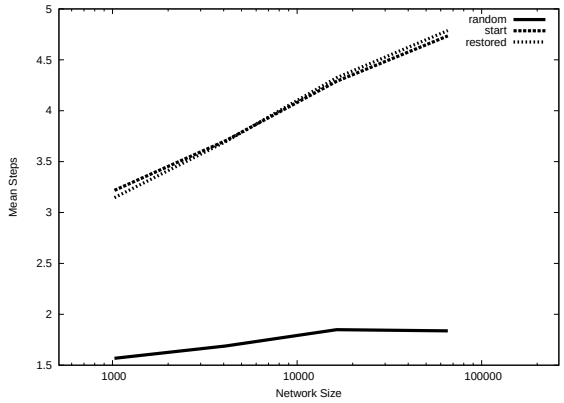

Figure 3.10: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Mean number of steps of successful queries when failing at dead-end nodes.

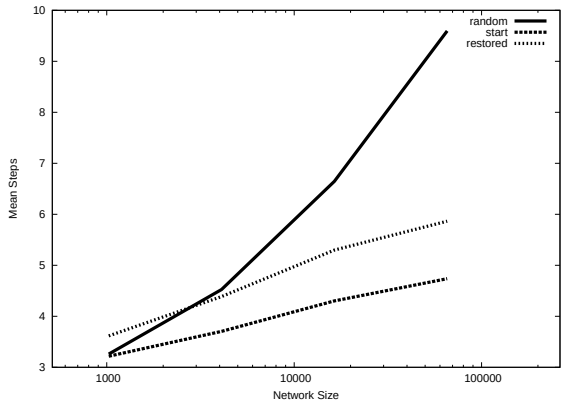

Figure 3.11: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Mean number of steps of queries when they are allowed to use local connections.

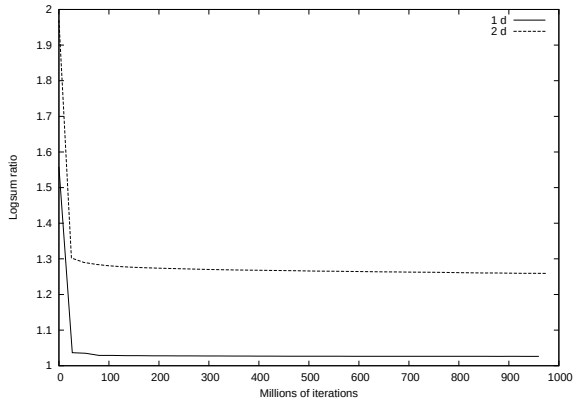

Figure 3.12: The target function of the optimization (log sum of shortcut distances) as the algorithm progresses. The graphs have 10000 nodes with edges generated using the ideal model. The values are normalized by dividing by the log sum of the original graph: it can be seen that we come much closer to restoring this value in 1 dimension.

mensional case, but it performs better than against the one dimensional algorithm did on the graphs generated from non-ideal models. See Figures 3.9 to 3.11 for some of the data.

It seems that the algorithm proposed here simply does not function as well in the two-dimensional case. In Figure 3.12 the sum of the logarithms of the shortcut distances for a graph is plotted as the optimization is run for a very large number of iterations. It indicates that results in two-dimensions cannot be fixed by simply running more iterations, in fact, it seems like it fails to converge to one completely.

Graphs generated according to the two dimensional model were also given to the one dimensional algorithm, and vice versa. We found that data from either model was best analyzed by fitting it against a base graph of the same dimension - but the two dimensional method actually did slightly better on one-dimensional data than its own. For example at a network size of 4096, we were able to restore a success rate of 0.670 when failing at dead-ends using the two dimensional method for one dimensional data, but only 0.650 on data from the two dimensional model. This indicates that the worse performance in two dimensions may be largely due to Kleinberg's model in higher dimensions being more difficult to fit correctly.

3.7.3 Real World Data

We treated the real world data in the same way as the simulated graphs. 2000 and 4000 vertex subgraphs were generated using the procedure defined above, the nodes were given random positions in a base graph, and then $6000n$ iterations of the Metropolis-Hastings algorithm was performed. We tried embedding the graph both in the one dimensional case (circle) and two (torus). In one dimension, the results were as follows:

Size	2000	4000
Mean degree	64.6	46.4
F Success	0.609	0.341
F Steps	2.99	3.24
C Succ	0.981	0.798
C Steps	13.4	26.0
LC Steps	4.58	7.21

Here "F Success/Steps" denotes the values when we fail upon hitting a dead end, "C Succ/Steps" when we continue and "LC steps" is the mean number of steps for queries that use the local connections at dead ends.

The data was also tested using two-dimensional coordinates and distance. The results are rather similar, with some of the tests performing a little bit better, and some (notably the success rate when failing on dead ends) considerably worse.

Size	2000	4000
F Success	0.494	0.323
F Steps	2.706	3.100
C Succ	0.984	0.874
C Steps	13.116	22.468
LC Steps	3.920	5.331

It perhaps surprising that using two dimensions does not work better, since one would expect the greater freedom of the two dimensional assignment to fit better with the real dynamics of social networks (people are, after all, not actually one a circle). The trend was similar with three-dimensional coordinates, which led to success rates of 0.42 and 0.26 respectively for the large and small graphs when failing at dead-ends, but similar results to the others when continuing. As can be seen from simulations above, the algorithm does not seem to perform very well in general in higher dimensions, and this may well be the culprit.[6]

[6]There is a general perception that the two-dimensional case represents reality, since peoples geographical whereabouts are two-dimensional. We find this reasoning somewhat specious. The true metric of what makes two people closer (that is, more likely to know one another) is probably much more complicated

The two thousand node case has about the same degree as the simulated data from the graphs above, so we can compare the performance. From this we can see that the "web of trust" does not nearly match the data from the ideal model in any category. It does, however, seem to show better performance than the uniform matchings in some cases - most notably the crucial criteria of success rate when dropping at dead ends.

To look at the 4000 nodes case, the mean degree is considerably less than the experiments presented below, and it the results are unsurprisingly worse. In this case however, the dataset does have a lot of nodes with only a few neighbors, and it is easy to understand it is difficult for the algorithm to place those correctly.

At first glance, these results may seem rather negative, but we believe there is cause for cautious optimism. For one thing, success rates when searching in real social networks have always been rather low. In [26], when routing using actual geographic data, only 13% of the queries were successful. They used a considerably larger and less dense graph than ours, but on the other hand they required only that the query would reach the same town as the target. [2] showed similar results when attempting to route among university students. Real world Milgram type experiments have never had high success rates either: Milgram originally got only around 20% of his queries through to the destination, and a more recent replication of the experiment using the Internet [13] had as few as 1.5% of queries succeed.

Moreover, there have not been, to the authors knowledge, any previously suggested methods for routing when giving nothing but a graph. Methods suggested earlier for searching in such situations have been to either walk randomly, or send queries to nodes of high degree. With this in mind, even limited success may find practical applications.

---

than just geography (the author of this article is, for instance, perhaps more likely to know somebody working in his field in New Zealand, than a random person a town or two away). In any case, there is a trade-off between the realism of a certain base graph, and how well the optimization seems to function, which may well motivate less realistic choices.

3.8 Distributed Implementation and Practical Applications

The proposed model can easily be implemented in a distributed fashion. The selection kernel used in the simulations above is not decentralized, in that it involves picking two nodes x and y uniformly from the set. However, the alternative method is that nodes start random walks of some length at random times, and then propose to switch with the node at which the walk terminates. Simulating this with random walks of length $\log_2(n)/2$ (the log scaling motivated by the presumed log scaling of the graphs diameter) did not perform measurably worse in simulations than a uniform choice (nor on the collected data in the last section)[7]. For example, in a graph of 64,000 nodes generated with the ideal distribution, we get(with the tests as described above):

Test	Success Rate	Mean Steps
Fail	0.668	4.059
Continue	0.996	6.039
Base Graph	1.0	4.33

Once the nodes x and y have established contact (presumably via a communication tunnel through other nodes), they require only local data in order to calculate the value in (3.3) and decide whether to switch positions. The amount of network traffic for this would be relatively large, but not prohibitively so.

In a fully decentralized setting, the algorithm could be run with the nodes independently joining the network, and connecting to their neighbors in the shortcut graph. They then choose a position randomly from a continuum, and start initiating exchange queries at random intervals. It is hard to say when such a system could terminate, but nodes could, for example, start increasing the intervals between exchange queries after they have been in the network long. As long as some switching is going on, of course, a nodes position

[7]The most direct decentralized method, that nodes only ever switch positions with their neighbors, did not work well in simulation.

would not be static, but at any particular time they may be reachable.

The perhaps most direct application for this kind of process, when the base graph is a social network between people, is an overlay network on the Internet, where friends connect only to each other, and then wish to be able to communicate with people throughout the network. Such networks, because they are difficult to analyze, have been called "Darknets", and sometimes also "Friend-to-Friend" (F2F) networks.

3.9 Conclusion

We have approached a largely unexplored question regarding how to apply small-world models to actually find greedy paths when only a graph is presented. The method we have chosen to explore is a direct application of the well known Metropolis-Hastings algorithm, and it yields satisfactory results in many cases. While not always able to restore the desired behavior, it leads to better search performance than can be expected from simpler methods like random searches.

Much work remains to be done in the area. The algorithm depends, at its heart, on selecting nodes who attempt to switch positions with each other in the base graph. Currently the nodes that attempt to switch are chosen uniformly at random, but better performance should be possible with smarter choice of whom to exchange with. Something closer to the Gibbs sampler, where the selection kernel is the distribution of the sites being updated, conditioned on the current value of those that are not, might perhaps yield better results.

Taking a step back, one also needs to evaluate other methods of stochastic optimization, to see if they can be applicable and yield a better result. No other such method, to the author's knowledge, applies as directly to the situation as the Metropolis-Hastings/simulated annealing approach used here, but it may be possible to adapt other types of evolutionary methods to it.

Also, all the methods explored here are based on the geographic models that Kleinberg used in his original small-world paper [23].

His later work on the dynamics of information [24] (and also [36]), revisited the problem with hierarchical models, and finally a group based abstraction covering both. It is possible to apply the same techniques discussed below to the other models, and it is an interesting question (that goes to the heart of how social networks are formed) whether the results would be better for real world data.

The final question, whether this can be used successfully to route in real life social networks is not conclusively answered. The results on the limited datasets we have tried have shown that while it does work to some respect, the results are far from what could be hoped for. Attempting to apply this method, or any derivations thereof, to other real life social networks is an important future task.

Bibliography

[1] A. Abdul-Rahman. The pgp trust model. *EDI-Forum: the Journal of Electronic Commerce*, 1997.

[2] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27:187–203, 2005.

[3] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman. Search in power-law networks. *Physical Review E*, 64 (46135), 2001.

[4] N. Alon, J.H. Spencer, and P. Erdös. *The Probabilistic Method*. Wiley, 1992.

[5] L. Barriere, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In *Proceedings of the 15th International Symposium on Distributed Computing, DISC'01*, 2001.

[6] B. Bollobas. The diameter of random graphs. *Trans. Amer. Math. Soc.*, 267:41–52, 1981.

[7] B. Bollobas. *Random Graphs*. Academic Press, 1985.

[8] B. Bollobas and F. Chung. The diameter of a cycle plus a random matching. *SIAM Journal on Discrete Mathematics*, 1:328–333, 1988.

[9] I. Clarke, T. Hong, S. Miller, O. Sandberg, and B. Wiley. Protecting free expression online with freenet. *IEEE Internet Computing*, 6:40–49, 2002.

[10] I. Clarke, T. W. Hong, O. Sandberg, and B. Wiley. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, pages 311–320, 2000.

[11] A. Clauset and C. Moore. How do networks become navigable? *Preprint*, 2003.

[12] I. de S. Pool and M. Kochen. Contacts and influence. *Social Networks*, 1:1–48, 1978.

[13] P. S. Dodds, M. Roby, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301:827, 2003.

[14] R. Durrett. Random graph dynamics. book draft, 2005.

[15] David Eppstein and Joseph Yannkae Wang. A steady state model for graph power laws. In *2nd International Workshop on Web Dynamics*, May 2002.

[16] P. Erdös and A. Renyi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[17] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. *Theory of Computer Science*, 1:237–267, 1978.

[18] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *INFOCOM*, 2004.

[19] O. Häggström. *Finite Markov Chains and Algorithmic Applications*. Number 52 in London Mathematical Society Student Texts. Cambridge University Press, 2002.

[20] O. Häggström. *Slumpens Skördar: Strövtåg i sannolikhetsteorin*. Studentlitteratur, 2004.

[21] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[22] S Janson, T Luczak, and A Rucinski. *Random Graphs*. Wiley, 2000.

[23] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[24] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS) 14*, 2001.

[25] J. Kleinfeld. Could it be a big world after all? *Society*, 39, 2002.

[26] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geograph routing in social networks. In *Proceedings of the National Academy of Science*, volume 102, pages 11623–11628, 2005.

[27] C. Martel and V. Nguyen. Analyzing kleinberg's (and other) small-world models. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 179–188, New York, NY, USA, 2004. ACM Press.

[28] S. Milgram. The small world problem. *Psychology Today*, 1:61, 1961.

[29] M. Newman. Models of the small world: A review. *Journal of Statistical Physics*, 101:819–841, 2000.

[30] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[31] M. Newman and D. Watts. Renormalization group analysis of the small-world network model. *Phys. Lett. A*, 263:341–346, 1999.

[32] O. Sandberg and I. Clarke. An evolving model for small world neighbor selection. draft, 2005.

[33] R. Solomonoff and Rapoport A. Connectivity of random nets. *Bull. Math. Biophys.*, 13:107–117, 1951.

[34] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[35] D.J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.

[36] D.J. Watts, P. Dodds, and M. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[37] D.J. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393:440, 1998.

[38] B. Yu and M. Singh. Search in referral network. In *Proceedings of AAMAS Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, 2002.

[39] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *Proc. IEEE Infocom*, 2002.