THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

# The Structure and Dynamics of Navigable Networks

**Oskar Sandberg**

CHALMERS | GÖTEBORG UNIVERSITY

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and Göteborg University
Göteborg, Sweden 2007

The Structure and Dynamics of Navigable Networks
Oskar Sandberg
ISBN 978-91-628-7308-0

Photo of Paul Erdös on page 15 taken in 1989 by George Csicsery for his film ”N is a Number: A Portrait of Paul Erdös” (1993). ©1993 All Rights Reserved. www.zalafilms.com. Used with permission.
Photo of Stanley Milgram on page 11 ©Al Satterwhite. www.alsatterwhite.com. Used with permission.
Poem by Berton Braley on Page v first published in “Adventure Magazine”, 1921. It is now in the Public Domain.

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

Typeset with LaTeX.
Printed in Göteborg, Sweden 2007

## Abstract

The notion that the world's social network is tightly connected and that any two people can be linked by a short chain of friends, known as the small-world phenomenon, has long been a subject of interest. Famously, the psychologist Stanley Milgram performed an experiment to investigate how long it would take for letters to cross the United States going from friend to friend. The results seemed to confirm that the small-world phenomenon is real. More recently it has been shown by Jon Kleinberg that for it to be possible to efficiently search for paths in a random graph with only local knowledge, as Milgram's subjects did, the graph must have certain special properties. Such graphs, known as navigable networks, must combine structured and random elements.

We start out from Kleinberg's work and explore a series of results and ideas related to navigable graphs. Firstly, in two different models, we explore possible dynamics which could explain the emergence of navigability. The first model shows how a graph can adapt to navigability if one allows the edges to be re-wired based on the results of past searches, while the second model shows that navigability arises directly if vertices cluster with respect to two independent spaces in a certain, intuitively clear, manner.

The final two works take different perspectives. The first of these views the small world as a random graph and looks at its connectivity properties in the classical sense. Our results bridge a gap between previously understood results from random graph theory, and results from the study of percolation. In the second, final, work 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. The problem is motivated by applications to computer networks, and our method is tested on real world data.

**Keywords:** random graphs, small-world, navigable, networks, routing, clustering

**Subject Classification (MCS):** 05C80, 60J10, 60J22, 68R10, 68W15, 68W20, 91D30

## Acknowledgments

My debt in gratitude to all the people – family, friends, advisers, teachers, colleagues, and others – who have helped me reach this point is greater than I could ever possibly repay. One could not be given a life more charmed with opportunity than that which you have granted me. This work is the least of what could be expected from somebody with such fortune: for anything it may lack, I have no excuse.

---

The reason I never can quit the road
Is a reason that's plain and clear.
It's because no matter where I may stop
And whether it's far or near
Ther's a place beyond the place I am,
Wherever I may be at,
And then beyond is a place beyond
And the world beyond all that!

And as long as a man has eyes to see
And a brain that wants to know,
I figure ther's things he's bound to miss
If he doesn't go on and go;
For there's always a place beyond the place
I happen to hang my hat,
And another place beyond that place
And the world beyond all that!

There's some folks stay in a single spot
Or a town of which they're fond,
And never worry a little bit
At the thought of a place beyond;
But the place beyond the place beyond
Won't never let me rest
For there's a sort of a kind of urge
That's burnin' within my breast –

To go an' go till the end of life,
An' when I've left it flat,
Go on beyond the place beyond;
And the universe after that!

"A Little Further" by Berton Braley

# Contents

# Chapter 1

# Introduction

While it has been brought together as a single text, this thesis actually consists of a collection of works regarding random graphs. In particular, it concerns models where graphs have a geometrical structure, meaning that the existence of an edge between two vertices depends on the distance between them. The main subject of study about these models is that of navigability: is it possible to find paths between vertices without global knowledge of the graph realization?

## 1.1 Summary of Contents

### Chapter 2: Background

We start with a summary survey of the so called “small-world phenomenon” and the mathematical models it has inspired. The story begins with Milgram’s small-world experiment [48], which showed that people are able to do a surprisingly good job at finding paths through the world’s social network. A brief survey of the mathematical models inspired by these ideas is given.

Most importantly, we emphasize the small world models of Jon Kleinberg [39], and his results about decentralized search. Kleinberg modeled the world’s social network as a graph embedded in a two-dimensional space, and let the probability of there being an edge between two vertices depend on the distance between them. In such models, he proved, decentralized search methods are only efficient if the relation between probability and distance takes a particular form. Specifically, if it is a power-law, there is only one exponent in the power-law relation which results in what has later been dubbed a navigable network.

## Chapter 3: Destination Sampling

A question left open by Kleinberg and previous work in the area is why one should expect real world networks to have navigable characteristics. Since Milgram's experiment was performed on the world's social network, it seems that it, at least to some extent, is navigable, and yet the mathematics tells us that this property only arises when networks have a very special form.

In this paper we explore an algorithm which randomly "re-wires" a graph. The algorithm works by performing many searches between random vertices in the graph, and after each successful search updating the edges of some of the vertices the search has passed through. The result of the algorithm is a graph-valued Markov chain, whose stationary distribution obeys the following property:

> The distribution of edges coming out of a vertex is the same as the distribution of destinations of searches that reach that vertex.

This property is self-optimizing: if a vertex sees many queries for a particular destination, it makes sense that it should have a high probability of linking there (and vice versa). Under an additional independence assumption (which unfortunately does not hold for the stationary distribution of the algorithm), we show that graphs obeying this property are navigable in certain discrete and continuous settings.

*This chapter is to be published as "Neighbor Selection and Hitting Probability in Small-World Graphs" in the Annals of Applied Probability, 2007.*

### Further work with Olof Mogren

In his master's thesis in computer science [49], Olof Mogren has continued simulation studies of the rewiring algorithm. In particular, he has studied the more realistic situation where vertices are not evenly distributed in the world: some areas have many nearby vertices, while others are more sparsely populated. Liben-Nowell et al. [44] have previously shown that direct addition of edges, using the formulas given by Kleinberg for the grid, will not work (in the sense of making the resulting graph navigable) in these cases, but the edge probabilities must respect the population density.

Mogren's simulation results indicate that the updating process discussed in the previous section does respect population density and will make graphs navigable even when the underlying population has an uneven distribution. The methods were tested both on simulated population models, and the actual population distribution of contemporary Sweden (see Figure 1.1).

### Chapter 4: Double Clustering

In our second attempt to explain the emergence of navigable networks, we present a different, yet perhaps even more compelling construction. The new method, which we have dubbed "double clustering", assigns each vertex a position in two independent spaces. In the social network context, these two spaces can be seen as the physical place where a person lives, and a more metaphoric space of "interests" (where people work, what their hobbies are, and so on).

We then show that navigability arises when each vertex has edges to its best peers with respect to both spaces in the following sense: a vertex $x$ will have an edge to another vertex $y$ if there is no third vertex which is closer to $x$ than $y$ in *both* spaces. Thus $x$ will always have a link to the very closest vertex in either space, but to the second closest only if it is closer in the other space than the very closest, and so on. Again in the social network context, a vertex will link to any vertex which is more interesting than all vertices that live closer to him (conversely, it will not link to a vertex which is less interesting than one which is closer).

Graphs constructed in this manner turn out to navigable with respect to both spaces, which is a compelling property since in the Milgram experiments people could route based both on the target's location and on his occupation. Mathematically, we are able to prove that double clustering graphs are navigable in several cases. We conjecture that the construction works under very general situations.

*This chapter is a preprint of an article that has been submitted for publication.*

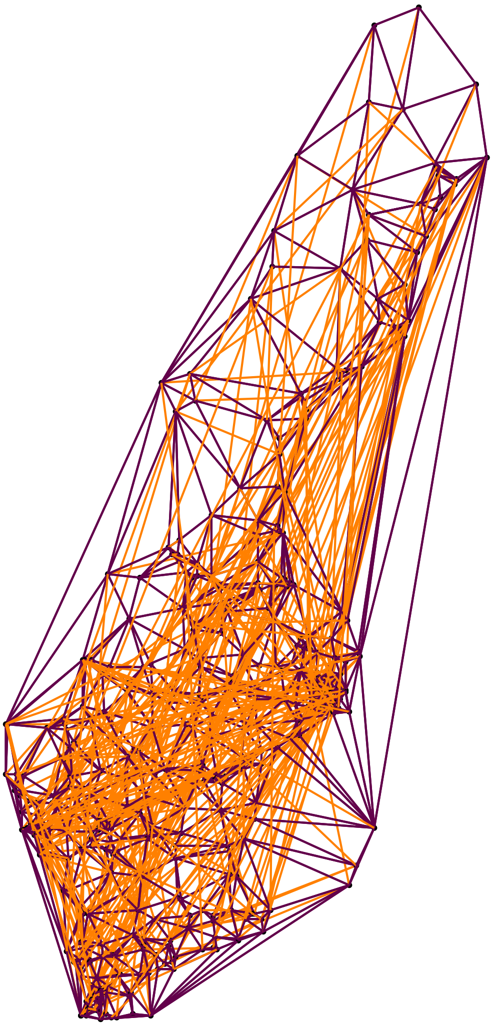


Figure 1.1: A simulated social network reflecting the population distribution of Sweden. Vertices (representing people) are distributed according to data acquired from Statistics Sweden, after which the edges (friendships) have been created by the methods described in Chapter 3. The pictured network is navigable. Image by Olof Mogren, see [49].

## Chapter 5: Phase Transitions of Partially Structured Graphs

This takes a more traditional approach to the analysis of random graphs. In particular, we take an interest in the continuum of models between traditional random graphs of the Erdős-Rényi type [26] [10], where there is no underlying structure, and percolation models [34], where the possible edges are dictated exactly by a geometry. The navigable graphs described above belong in-between of these models. When edges are a just "a little less" (in a certain sense) dependent on the vertex distance than in the navigable case, the result may be analyzed as a traditional random graph. When it is just a "a little more" the model falls into previously analyzed percolation regimes. The navigable graphs thus form a critical boundary, however, we prove that they still undergo a similar phase transition where a giant connected cluster arises.

*This chapter is a preprint of an article that has been submitted for publication.*

## Chapter 6: Distributed Routing

The final paper takes a more pragmatic perspective. It asks the question of how to search in a completely oblivious environment. Specifically, Kleinberg's models and the navigability results discussed earlier in the thesis all assume that vertices know the positions of their neighbors, giving them the possibility of answering questions of the form, "Which of my neighbors is closest to the destination of the search?" At the very least, vertices must know their own position. In this paper, we develop a way of routing without access to any such information, which works by using a distributed Markov chain Monte-Carlo method [35] to assign positions to vertices in an imaginary space, which can then be used for routing. We test our algorithm using simulation on networks generated to be navigable, as well as real world data taken from the world's social network.

*This paper was published as "Distributed Routing in a Small World" in the refereed proceedings of the SIAM ALENEX Workshop on experimental algorithms in January 2006.*

**Further work with Vilhelm Verendel**

In his masters thesis in complex adaptive systems [59], Vilhelm Verendel has studied alternative MCMC methods for assigning positions to the vertices, which lead to faster convergence toward a navigable equilibrium. Verendel's results show that large improvements can be made on the number of steps needed before searching becomes possible can be reduced dramatically by appropriate tuning the Metropolis chains sample distribution.

## 1.2 Future Directions of Research

The field of navigable networks is young, and much work remains to be done. In his 2006 ICM survey [41], Kleinberg listed ten areas of future research: six open problems and four more general questions. Two of the latter relate to work found in this thesis.

The largest part of this thesis relates to Kleinberg's seventh problem, understanding the evolution of navigable networks. While we feel, in particular with Chapter 4, that we have made a solid contribution to the understanding of this issue, it is by no means solved. A truly credible model of small-world evolution would need to include much more of the complicated interplay between members of the network observed in the real world – it is unlikely that a complete such model could ever be formulated, but improvements on what is presented here are possible.

More specifically, each of the works in this thesis leaves open many questions for future research. In Chapter 3 the main question of proving rigorously that the rewiring algorithm leads to navigable graphs is left open (though we are convinced that it does). Even the independent models discussed are limited to a few base graphs, while we believe the algorithm will lead to navigability in just about any situation. Similarly, we conjecture that the theorems regarding the double clustering graph in Chapter 4 hold in great generality, but have only proved them for a limited set of models.

Chapter 5 equally leaves many questions open. In terms of connectivity, the critical density of the long-range percolation models discussed is not known. For the particular case in which we prove the existence of a giant cluster, one may also wish to prove diameter results, as a greater understanding of the smaller clusters. In general, any of the multitude of questions that have been answered about Erdős-Rényi graphs in the last fifty years can also be asked about these models.

Finally, Chapter 6 is an experimental study, and the question of what can be proved is left completely open: in particular, results about convergence rate would be of great interest. Besides the work of Verendel [59] mentioned above, further experimental work has already been done by Dell'Amico [19] who attained positive results when replacing our MCMC approach to vertex placement with algorithms previously devised for graph-drawing. Evans et al. [27] have studied security issues related to deploying our algorithm in possibly hostile environments. In both these subjects, and many others, there is plenty of potential for further advances.

# Chapter 2

# Background

## 2.1 Milgram's Small-World Experiment

In 1967 Stanley Milgram set out to measure the size of the world. Not its physical size, but rather its social size: how far apart are we all if we count distance not in terms of kilometers and miles, but rather in terms of friendships and handshakes?

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 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's a small world" has become part of everyday speech.

In order to explore this matter further, Milgram proposed a simple experiment. Starting with volunteers from a city in the American midwest, he gave them packages intended to be forwarded to a previously chosen target in New England by mail. The experiment had a catch however: recipients of the package could not send it to just anybody, nor just directly to the final target, but had to send it to somebody with which they were acquainted (defined, for the experiment, as somebody with whom they were on first name basis). The paths that the packages traveled were thus chains of friends from the initial volunteers to the recipient [48] [58].

Milgram and his associates conducted the experiment several times with different starting groups (from Wichita in Kansas, Omaha in

Nebraska, and, for comparison, Boston, Massachusetts) as well as several targets (a stockbroker living outside Boston and the wife of a Divinities 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. In one of the studies the media number of steps six, a number that so caught the popular imagination that the term "six degrees of separation" has become part of our culture.

With time, the small-world phenomenon has reached beyond psychology and sociology. Especially in recent years, 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 [9] [18], continued through the well known small-world models of Watts and Strogatz [62] [60], and, importantly for the present work, the work of Jon Kleinberg [39] [40] about network search.

Milgram's experiment is the starting point of almost all small-world discussion, and few papers come without a reference to the 1967 article describing it (published in Psychology Today, a popular magazine rather than a scientific journal). 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 been accepted as a fact 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. The biggest problem, it turns out, with carrying out Milgram's type of small-world experiment is the number of completed chains. 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 studies, 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. The numbers that Milgram reported, including the famous six degrees, are only for the chains that were completed – nothing is known about how many steps it might have taken had those chains that failed not been abandoned.

In an article by psychologist Judith Kleinfeld [42], a large number of other problems with the study are cited. The subjects were selected in ways that made them more similar than truly random subjects, and several replications of the experiment, where the success rate was considered 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.

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[1] have solicited volunteers to start chains aimed at reaching 18 preselected targets in 13 countries [20]. 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.

**Stanley Milgram** (August 15, 1933 – December 20, 1984) was an American social psychologist, and one of the most fascinating, and controversial, experimenters in his field. Besides the small-world experiment, he is even more famous for his work on obedience. In his obedience experiments, he convinced the subjects that they were actually assisting, and that an actor was the actual subject. The pretend subject would be given a test, and the volunteers were asked to administer increasingly intense (though actually fake) electric shocks for each wrong answer. The frightening conclusion was that in some situations 65% of the volunteers were willing to administer the full 450 volts even as the "subject" screamed in pain.

**Biography:** Thomas Blass, *The Man Who Shocked the World*, Basic Books, 2004

Photo ©Al Satterwhite (www.alsatterwhite.com)

While the large number of chains seems promising, the observed success rates make those achieved by Milgram seem stellar. Of the 24,163

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

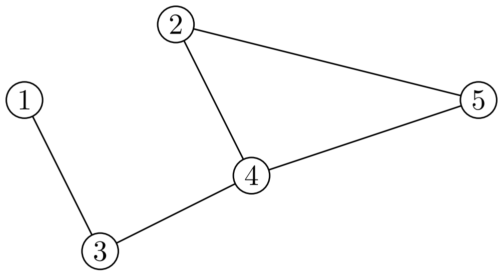
chains started, only 384 (a little over 1.5 percent) actually reached their intended destinations. This can be considered to support Milgram's hypothesis that the success rate depends on the perceived value of the parcel: few people, if any, will see much value in an Internet chain letter. For the completed chains, the average number 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, 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 was 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.

The advantage of the Internet based experiment over previous, letter based ones, was, however, that the use of the computer network allowed the Columbia team to track the chains at every step. They could see where and at what rate queries terminated. The findings were that the number of people 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 cause attrition of the chains.

If one assumes that chains really are dropped independently at each step, one can calculate the true distribution of chain lengths mathematically. Using such a method, the Columbia team arrived at an adjusted median value of seven. 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 indeed is small, at least by some measurements. But it is also a much more complicated world than a single number or experiment can explain. The mythical “six degrees” of popular culture are likely to remain just that.



The concept of a structure of points and the lines connecting them is ubiquitous in many scientific fields. It is called a "graph" in mathematics, with the points denoted as "vertices", and the lines as "edges". In computer science it is usually called a "network", with "nodes" and "links" or "connections". Physicists often speak of systems of "sites" and "bonds". Finally, in sociology it is often referred to as a network of "people", or "actors", and "contacts", "friendships", or "ties".

## 2.2 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 joint paper by political scientist Ithiel de Sola Pool and mathematician Manfred Kochen, but due to a lack of progress it was not published until 1978 [18] (it was actually reading about de Sola Pool's ideas that inspired Milgram to perform his experiment [8]). Since then a lot of work has been done, especially through computer simulation, but many theoretical questions remain open.

Seen from a mathematical perspective, the small-world phenomenon is a problem of graph theory (see box on page 13). 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 (the so called geodesic distance). One wants to bound the mean such distance over every pair of vertices, or, ideally, the maximum distance between any pair, also known as the graph's diameter.

Since social networks are formed through chaotic unobserved processes, the only sensible way to model them is to select the edges randomly. The study of such random graphs was initiated by Paul Erdős and Alfred Rényi in 1959 [26]. The model known as $\mathcal{G}(n,p)$, usually called the Erdős-Rényi model[2], is constructed by taking a set $V$ of vertices, and connecting each disjoint pair of vertices with probability $p$. These graphs have many interesting properties which have been the source of much study in probabilistic combinatorics [5] [10] [37]. In particular, it is known that when $p \leq 1/n$, the graph becomes one of small connected "islands" (known as components) of vertices with no edges between them, while for sufficiently large $p$ there is one giant component which dominates all the others (see also Chapter 5). The diameter of this large component is a logarithm of its size, meaning that for its inhabitants the reachable world is indeed small.

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, their other properties do not coincide with most observed networks. One oft-given example of this is that of clustering. A graph is clustered if two vertices sharing a common neighbor are more likely to be connected than two vertices chosen at random. While such behavior is very common in the real world (think, for example, about your own friends), it is not the case in the above model. In $\mathcal{G}(n,p)$ all vertex pairs are independently connected with the same probability, regardless of who else they know.

Formally, one defines the clustering coefficient, $c_{\text{clust}}$ of a (random) graph as the average (expected) portion of a vertex's neighbors which are also connected to each other. Clearly $c_{\text{clust}} = 1$ for a complete graph (one where all pairs of vertices are connected), $c_{\text{clust}} = 0$ for trees, and $c_{\text{clust}} = p$ for random graphs of the type discussed above.

Non-complete graphs with higher clustering coefficients can be constructed, most easily through so called nearest-neighbor graphs: start with a regular lattice of size $n$, which in 1 dimension is just a line of

[2] This model is actually due to Gilbert [32] - Erdős and Rényi's original model $\mathcal{G}(n,M)$ meant choosing a graph uniformly from all possible graphs with $n$ vertices and $M$ edges. When $M \approx p\binom{n}{2}$ the two models are, however, very similar. Solomonoff and Rapoport [57] had presented similar ideas outside mathematics earlier.

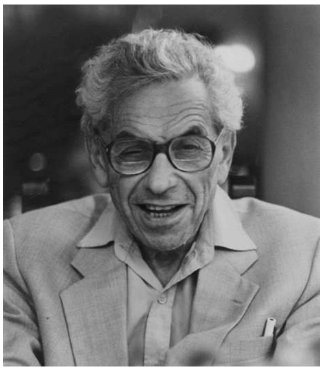
vertices, and add edges from each vertex to the $k$ nearest in the lattice. In such a graph can easily be seen (see [51]) that, if $k < \frac{2}{3}n$:

$$c_{\text{clust}} = \frac{3(k-2d)}{4(k-d)}$$

where $d$ is the dimension of the lattice.

The modern idea of a small-world graph, as introduced by Watts and Strogatz [61], is one which displays both the small diameter of the random graph, and the heavy clustering of organized nearest-neighbor graphs. For this to make sense, one needs to limit oneself to sparse graphs – those where the number of edges attached to each vertex (the vertex degree) grows slowly or not at all compared to the size of the graph. Even then the terms "small diameter" and "heavy clustering" are rather ambiguous, and have often been used somewhat loosely.

Roughly, a "small diameter" is understood to mean that the diameter should grow logarithmically (or at most polylogarithmically) compared to the size of the graph. "Heavy clustering" usually means that the clustering coefficient should not fall considerably when the graph grows but the number of neighbors of each vertex stays constant (in $\mathcal{G}(n,p)$ keeping the number of neighbors constant means



**Paul Erdős** (March 26, 1913 - September 20, 1996), was an immensely prolific Hungarian-born mathematician. He lived his life as a mathematical nomad, appearing at the offices and homes of fellow mathematicians and declaring his willingness to collaborate with the words "my brain is open".

Erdős had so many collaborators (511), that in the graph formed by joining with an edge any two mathematicians who have worked together, the small-world phenomenon is most often expressed in terms of people's "Erdős number" - the number of steps from them to the great Hungarian.

**Biography:** Paul Hoffman, *The Man Who Loved Only Numbers*, Hyperion 1999.

Photo ©George Csicsery (www.zalafilms.com) that $p = c/n$ for some $c > 0$).

Watts and Strogatz 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 similar to the random graphs of Erdős and Rényi. 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. A more approachable model is to take a cycle and add a random matching between the vertices. Bollobás and Chung [11] proved already in the 1980's that the diameter of such graphs grows logarithmically with high probability (in fact, they give an asymptotic formula for the diameter including the constants). Adding further nearest-neighbor edges to such a graph can not increase the diameter, so it is easy to construct a provably "small-world" graph in this manner.

With this result in mind, it isn't difficult to imagine that the same thing holds for similar mixes of structured and unstructured graphs. There is a dearth of rigorous results, however, with most published results relying on simulation or so called mean field approximations. Summaries of existing results, as viewed by physicists, can be found in the reviews by Newman [51][52]. A recent book by Durrett [24] also attempts to collect some of the more rigorous results available.

## 2.3 Navigation

While it is fair to say that the existence of random links can explain the small diameter of a network, there is a rather large gap between this conclusion, and what Milgram's experiment showed about the world's social network. Upon consideration, the fact that short paths exist between people is not nearly as surprising as the fact that people can find them. That is to say, in the experiment, each participant who receives the letter has to act as a *router* – deciding for himself who the next person in the chain should be. Each person has a very limited overview of the world: he knows, in essence, only who his own friends are, and maybe a little bit about his friends' friends, but rarely more. He thus has to choose who he routes to based on very little information: far less,

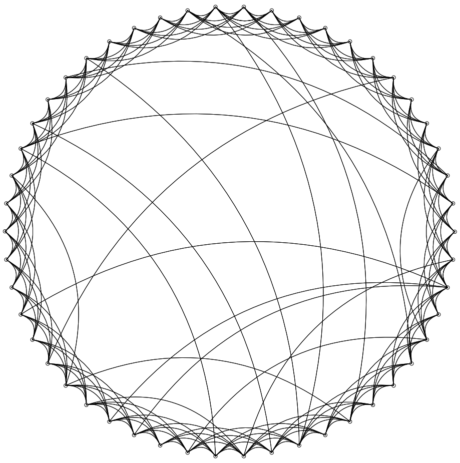


Figure 2.1: A small world constructed by adding random edges to a cycle.

for instance, than that used by traditional algorithms for finding the shortest path through a graph. How was it that the subjects were still so proficient at finding paths?

Jon Kleinberg tackled these questions in 2000. The result was a seminal paper [39] in which he showed that the previous small-world models could not explain this fact. If you take a grid-like graph, and add some random edges as in the models above, the diameter may become small, but it is simply not possible for any algorithm, working only with local knowledge of the graph, to efficiently find paths. The expected number of steps to find one point starting from another is lower bounded by a fractional power of the graph size, and is thus exponential in the diameter.

Moving on from this, Kleinberg defines a wider family of random graphs. Like the earlier small-world models, he starts with an underlying grid and adds shortcut edges between distant vertices, but this time allowing the probability that two vertices are connected by a long edge to depend on the distance between them in the grid. In particular, we let the probability that a vertex $x$ has a shortcut to a non-adjacent vertex

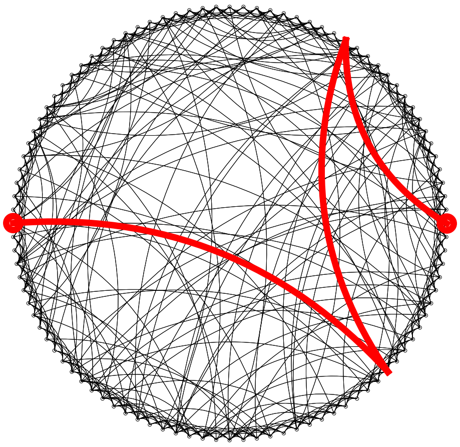


Figure 2.2: Even when a short path through a network exists, finding it can be difficult. How would each participant in the highlighted chain know where to send the message? If asked to forward a message towards a stockbroker in Boston, who would you send it to, and why?

$y$ have the form

$$\frac{d(x,y)^{-\alpha}}{h_{\alpha,n}} \tag{2.1}$$

where $d$ is the distance function between the vertices in the underlying grid, and $h_{\alpha,n}$ is a normalizer which makes sure these probabilities add to 1. In this family, he showed that $\alpha$ equal the dimension of the grid is the only case that allows for efficient navigation. The algorithm used in this case is a simple, *greedy*, procedure: in each step, the vertex sends the query on to that neighbor which is closest to the destination. This requires knowledge of nothing but the positions of the neighbors and the destination, yet can be proved (see below) to lead to paths of the order $(\log n)^2$.

This shows that the story behind the small world is not as simple as it may seem. While it is easy to produce graphs with low diameter and high clustering, *navigability*, as it has been called, requires something more. It is this fact that motivates the following work.

## 2.4 What About Degree Distribution?

At this point, the reader may accuse us of having neglected to mention another important aspect of the study of graphs as models of real life networks: namely the degree distribution. The degree of a vertex is the number of edges connected to it[3] (the number of friends somebody has). In models like the Erdős-Rényi graph, this tends to be sharply concentrated, in the sense that most vertices have approximately the average number of neighbors. In many real world networks, however, it tends to be much more skewed: some vertices have very many neighbors, while most have rather few (some people are very popular, while most are not).

In particular, if $N(x)$ is the number of neighbors that $x$ has, in Erdős-Rényi type constructions, then

$$\mathbf{P}(N(x) > k) \approx e^{-mk}$$

for some fixed $m > 0$, while for naturally occurring networks,

$$\mathbf{P}(N(x) > k) \approx k^{-\beta}$$

for some $\beta > 0$. The latter type of distribution is called a “power-law” or, especially by physicists, a “scale-free distribution”. The latter term has led to graphs having such degree distributions sometimes being referred to as “scale-free graphs”: a doubly unfortunate name since the graphs are not scale free in any meaningful sense, and since the degree distribution doesn’t characterize a random graph to begin with.

Neither the earlier small-world models, nor Kleinberg’s model, have particularly “interesting” degree distributions. For the Kleinberg model, the out-degree of each vertex is fixed, while the in-degree approaches a Poisson distribution as in the Erdős-Rényi graph.

The study and construction of graphs with power-law, or even arbitrary, degree distribution has been the overwhelmingly dominant enterprise within the field of random graphs in recent years, especially among probabilists (less so, perhaps, among computer scientists). Indeed, one could be forgiven for sometimes wondering if it has been forgotten that there is *anything* else to study about new random graph

[3] This should not be confused with the unrelated popular use of the word “degree” in “six degrees of separation”.

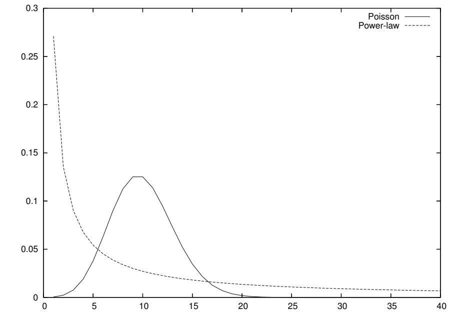


Figure 2.3: Frequency plots for a Poisson distribution vs a power-law, both with mean 10.

models. The number of papers published is too large, and growing too rapidly, for us to attempt to cite them here – we refer to a survey by Bollobás and Riordan [13] which is thorough but unfortunately a bit dated, and also a recent paper by Bollobás, Janson, and Riordan [12] which contains many references

In light of all this, we do admit some guilt. Degree distributions have already had their fair share – and perhaps more – of the attention. The type of models implied by navigability, where the randomness is mixed with a geographic structure, are interesting enough on their own.

One important question, however, is to what extent navigation, and Milgram's experiment in particular, can be considered without taking the degree distribution into account. One could be tempted to think that the routing used in such experiments was not, as in Kleinberg's model, attempting to "home in" on the destination by sending the letter to somebody as close it as possible, but rather that subjects utilized the skewed degree distribution by sending the letter to popular friends (who would be more likely to know the target).

We do not believe this to be the case. For one thing, it is clear that some other type of routing had to be employed, for the simple reason that a letter could not make it from Omaha to Boston in six steps simply by bouncing around between popular people (unless those people had hundreds of thousands of friends each). Also, the empirical evidence indicates otherwise: in the experiment of Dodds et al. [20] described above, participants were asked about their reason for choosing the person they forwarded the letter to. About 78% answered either some geographical similarity, or similarity of occupation, to the target while just 8% motivated their choice by the large number of friends of the person selected. They reference other studies with similar results. Adamic et al. showed in [3] that routing to the highest-degree neighbor can be efficient in power-law networks, yet when Adamic and Adar tried to employ this method on actual social network data [3], it did not work well, despite the degree distribution being skewed as expected. They achieved much better results using a method inspired by Kleinberg's work.

## 2.5 Why Navigation Matters

Before proceeding to present 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 the 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 in his experiment act as routers was simply necessity – nobody had global access to the world's social network, so no better way of routing was possible. This is less true today than it was then, and people have studied social networks in cases where the entire graph has been revealed, see [2] [44] and [54] (which is Chapter 6).

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 today this network of computers and wires is one of our most obvious examples of a real-world graph. 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. Complicated addressing and router systems are set up exactly to 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 [14] [15]. Similar work, drawing on Kleinberg's results, has been done by others [46] [64].

## 2.6 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. 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 [6], but below we will follow Kleinberg's original method rather than use their abstractions.

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 2.6.1.** *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 criterion 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 Chapter 6.

We now let the underlying grid be a closed cycle of $n$ vertices. If we identify the vertices with the numbers $0, 1, \ldots, n-1$, we have the distance formula:

$$d(x, y) = \min(|x-y|, n - |x-y|)$$

for geodesic distance along the cycle. The short edges are simply those to the vertices before and after each vertex in the cycle. Each vertex independently chooses the destination of an additional directed edge, which we henceforth refer to as its shortcut[5]. If we denote by $x \rightsquigarrow y$ the event that $x$ chooses $y$ as its shortcut destination, then

$$\mathbf{P}(x \rightsquigarrow y) = \frac{d(x, y)^{-\alpha}}{h_{\alpha,n}}$$

for some $\alpha$. 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 2.6.2.** (Kleinberg) *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 $\alpha$ but not on $n$.*

This, of course, leaves out the critical case where $\alpha = 1$, which we discuss below. Simply put, $\alpha < 1$ leads to too few shorter shortcuts, and

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

[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.

$\alpha > 1$ leads to too few longer shortcuts, while, as we shall see below, the critical middle is just right.

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

$$\begin{aligned} h_{n,\alpha} = 2\sum_{i=1}^{n/2} x^{-\alpha} &\geq \int_1^{n-1} x^{-\alpha} && (2.2) \\ &= (1-\alpha)^{-1}((n-1)^{1-\alpha} - 1) && (2.3) \\ &\geq \rho n^{1-\alpha} && (2.4) \end{aligned}$$

for some constant $\rho$ depending on $\alpha$ but not $n$.

Now we let $U$ be the set of vertices less than $n^\delta$ steps from $z$, where $\delta = (1-\alpha)/2$. Clearly $|U| \leq 2n^\delta$.

Let $A_i$ denote the event of finding a shortcut leading to $U$ in the $i$-th step.

$$\mathbf{P}(A_i) \leq \frac{|U|}{h_{n,\alpha}} \leq \frac{2n^\delta}{\rho n^{1-\alpha}}.$$

since $\mathbf{P}(x \rightsquigarrow y) \leq 1/h_{n,\alpha}$ for all $x$ and $y$.

If we now let $\lambda = \rho/8$ and $A = \bigcup_{i \leq \lambda n^\delta} A_i$ it follows that

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

Let $B$ be the event that starting point and target are separated by more than a quarter of the cycle, that is $B = \{d(Y,Z) > n/4\}$. Since we are choosing starting points uniformly

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

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

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

Let $T$ be the number of steps until we reach our target. The event $T \leq \lambda n^{\delta}$ cannot occur if $A^c \cap B$ does, since in order to reach the target in less then $\lambda n^{\delta}$ steps, we must at some point before then find a shortcut ending in $U$. In other words

$$\mathbf{P}(T \leq \lambda n^{\delta} \,|\, A^c \cap B) = 0$$

which implies

$$E_{\mathcal{A}}(T \,|\, A \cap B) \geq \lambda n^{\delta}.$$

By restriction it then holds that

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

Letting $k_1(\alpha) = \lambda/4$ gives the result.

**The case $\alpha < 1$:** We start by bounding the probability that a vertex $u$ has a shortcut destination $v$ that is more than $m$ steps away. Let $\epsilon = \alpha - 1 > 0$, and note that $h_{\alpha,n} \geq 1$.

$$\begin{aligned} \mathbf{P}(d(u,v) > m) &\leq 2 \sum_{j=m+1}^{N/2} j^{-\alpha} \\ &\leq 2 \int_{m}^{\infty} x^{-\alpha} dx \\ &= 2\epsilon^{-1} m^{-\epsilon} \end{aligned}$$

Now let $\gamma = 1/(1+\epsilon)$, and $A_i$ be the event that in the $i$-th step, we find a shortcut longer than $n^{\gamma}$. Also let $\mu = \min(\epsilon, 2)/8$, 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

$$\begin{aligned}\mathbf{P}(A) &\leq \sum_{i\leq\mu n^{\epsilon\gamma}} \mathbf{P}(A_i) \\ &\leq 2\mu n^{\epsilon\gamma}\epsilon^{-1}(n^{-\gamma})^{\epsilon} \\ &= 2\mu\epsilon^{-1} \leq \frac{1}{4}.\end{aligned}$$

Similarly to the first case, we let $B$ be the event that $d(Y,Z) > n/4$, 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 $\leq \mu n^{\epsilon\gamma+\gamma} = \mu n < n/4$. 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 $\mathcal{A}_G$ denote the following decentralized algorithm:

- At each step $X_t$, choose among the local neighbors and the shortcut the vertex $u$ 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.*

**Theorem 2.6.3.** (Kleinberg) *If $\alpha = 1$, then for all vertices $u$ and $v$*

$$E_{\mathcal{G}}(T \mid Y = u, Z = v) \leq k_3(\log n)^2.$$

*Proof.* Like before, we start by bounding the normalizer, $h_{n,1}$:

$$\begin{aligned}\sum_{v\neq u} d(u,v)^{-1} &\leq 2\sum_{i=1}^{n} i^{-1} \\ &\leq 2(1+\log n) \leq \kappa \log n\end{aligned}$$

for some constant $\kappa$. 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, i.e. 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

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

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 $\log_2(\log_2(n)) < j \leq \log_2(n)$:

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

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

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

□

# Chapter 3

# Destination Sampling

## 3.1 Introduction

### 3.1.1 Shortcut Graphs

Starting with the small-world model of Watts and Strogatz [62], 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 [51] [53]). 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 [39] 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, but also that if the probability of two vertices having a long-range edge between them had the correct relation to the distance between them in the grid, the *greedy routing* path-length between vertices was small as well. Greedy routing means, as the name implies, starting from one 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. Graphs where one can quickly route between two points using only local information at each step, as with greedy routing, are referred to as *navigable*.

Initially, we will stay in the one-dimensional translation invariant environment (that is, with the vertices arranged on a circle). Later sections extend some of the results to other classes of graphs. In general, we will call graphs of the type discussed *shortcut graphs* and use the shorter term *shortcut* for the long-range edges.

### 3.1.2 Contribution

While Kleinberg's results are important and have been a catalyst for much study, it is not fully understood how the rather arbitrary and precise threshold on the shortcut distribution might arise in practice. In this work, we present an alternative distributional requirement that associates the shortcut distribution with the hitting probabilities of walks under greedy routing. We study this in the canonical case of a single loop, and in a wider setting of graphs induced by the Voronoi tessellations of a Poisson process. We show that distributions that meet a certain criterion which we call being "balanced" have $O(\log^2 n)$ mean routing times, similar to the critical case in Kleinberg's model.

The relationship in this criterion naturally leads to a stepwise rewiring 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. Our analytic results cannot be applied directly in this case, because the stationary distribution has dependencies between the shortcuts at nearby vertices. However, we argue through heuristics and simulation that these dependencies in fact work in our favor, and that graphs generated by our algorithm can be efficiently navigated.

### 3.1.3 Previous Work

The roots of the recent work on navigable graphs are the papers by Jon Kleinberg [39] [38]. Further exposition is given in [6] [45] [47]. Continuum models similar to the ones discussed below have been introduced in [30], [21], and, in a more practical context, [44].

A very different algorithm that appears to produce navigable graphs has been independently proposed in [16], where it is tested by simulation. In [23] the emergence of navigable graphs is discussed in terms of a method for small-world construction without requiring an understanding of the geography, but the method developed is complicated and unnatural. An algorithm similar to that proposed below is present in Freenet [15] [14] [64] – the work below was in part inspired by attempts to place Freenet's algorithms in environments more conducive to analysis.

A recent survey of the field by Kleinberg is [41]. In the final section, he identifies the question of how small-world graphs arise as one of the central questions in the field.

## 3.2 Preliminaries

### 3.2.1 Decentralized Routing

The central problem in this area of research is that of *routing* through a graph with only limited knowledge of the graph itself. That is, given two vertices $x$ and $y$ in a (di)graph $G$, we want to find a path connecting $x$ and $y$. In general, the combinatorial problems of finding such a path, and finding the shortest such path, are well understood problems involving $\Theta(n)$ and $\Theta(n^2)$ steps respectively. The question becomes more interesting if we allow some (but not all) of the information about the graph to be known when determining the path. In particular, we know the distances between vertices as given by a function $d(x, y)$. With such a distance function, one may define a *decentralized algorithm* (following Kleinberg [39]) as an algorithm which, in each step, uses only information about the distances between vertices already seen in the route and the destination to decide where to go next.

**Definition 3.2.1.** *A* decentralized algorithm *for finding a path from a point $y$ to $z$ in a graph $G$ associated with a distance function $d : V \times V \mapsto \mathbb{R}$ is defined as follows.*

- *Let the $S_0 = \{y\}$.*
- *In step $k$, the algorithm chooses exactly one point in $N(S_{k-1})$ (the set of all neighbors in $G$ of points in $S_{k-1}$) and appends this point to create $S_k$. The choice of $x$ is a possibly random function of the subgraph of $G$ induced by $S_{k-1}$, as well as the distance of all the vertices in $N(S_{k-1})$ to each-other and to $z$ as given by $d$. In particular, it may not depend on the rest of $G$.*

- *The algorithm terminates in step $k$ if $z \in S_k$.*

The definition is inspired by the small world experiments [48] where people were enlisted to forward a letter to a stranger through friend-to-friend links. The people in the experiment knew something about the final recipient (typically where he lived and his occupation), so they could compare how “close” each acquaintance they considered sending the letter to was to to the target, but they had no global knowledge of the social network itself.

For a decentralized algorithm to be able to perform better than a random walk, it is necessary that $d(x, y)$ contains some information about the structure of the graph. The extreme of this is where $d(x, y)$ is the graph distance implied by $G$, the minimal distance from $x$ to $y$ in $G$, which we denote $d_G(x, y)$. In this case routing is trivial: proceeding in each step to the neighbor which is closest to $z$ will always find a minimal path. A more typical situation is that $d(x, y)$ gives some, but not complete, information regarding where to go. In particular, we shall say that $d(x, y)$ is *adapted for routing* in a graph $G$, if for any $z$ and $x \in V$, $x$ has a neighbor $y$ such that $d(y, z) < d(x, z)$. When such a distance measure exists, we can route to any point by always choosing such a $y$ as the next step, though the path thus found may be far from optimal.

The common situation is to let $H$ be a fixed graph, and $G$ created by randomly augmenting $H$ with further edges in order to create a semi-random graph. It is then trivially true that $d_H(x, y)$ is adapted for routing in $G$. The random edges need not be uniformly distributed, and indeed all the interesting cases arise when the probability of an edge being added between $x$ and $y$ depends on $d_H(x, y)$. Some independence is usually assumed, however, so that the edges previously seen in a route are independent of those in the future. We let $\ell(x, y)$ denote the probability of adding an edge from $x$ to $y$.

Given such a random augmentation of edges, the question arises whether a decentralized algorithm can be found which efficiently routes through a family of graphs. In particular, for a family for finite graphs of bounded degree that are indexed by size, is there a decentralized algorithm such that the expected number of steps of a route between two points is asymptotically small (by which we typically mean that it grows at most poly-logarithmically with the size).

In Kleinberg's original work [39], the underlying graph was $\mathbb{Z}_n^2$ (the family of finite two dimensional grids) with edges between adjacent vertices, making the $d(x, y)$ the $l^1$ metric (Manhattan distance). He proved that poly-logarithmic routing was possible if $\ell(x, z) = 1/(h_{n,\alpha} x^\alpha)$ with $\alpha = 2$ ($h_{n,\alpha}$ is the distribution's normalizer), but impossible for all other values of $\alpha$. Kleinberg's results also cover the same situation in $\mathbb{Z}_n^d$, in which case the single good value of $\alpha$ is exactly $d$. Similar analysis has been done by others: see e.g. Barriere et al. [6] for thorough analysis of the directed loop, and Duchon et al. [22] for a wider class of graph families. In all these cases (as well as in [40] [61] [44] [43]) it is found that efficient routing is possible when

$$\ell(x, y) \propto \frac{1}{\mathrm{Vol}(B_x(d(x, y)))} \tag{3.1}$$

where $B_x(r) = \{z : d(x, z) \leq r\}$, or some slight variation thereof. (We will use this notation for the ball, as well as $S_x(r)$ for its boundary throughout the paper.)

Similarly, it turns out that in all these cases, the decentralized algorithm necessary is simply *greedy routing*, which means choosing in each step the unexplored neighbor of the previously explored vertices which is closest to the destination. When $d(x, y)$ is adapter for routing, greedy routing strictly approaches the target with each step and is always successful. The nature of the greedy paths through augmented graphs is the main emphasis of this paper.

The following is a very coarse, obvious, upper bound on the routing time:

**Observation 3.2.2.** *If a distance function $d : V \times V \mapsto \mathbb{R}$ is adapted for routing in a graph in $G = (V, E)$ then greedy routing from $x$ to $z$ takes a number of steps which is at most the cardinality of $\{v \in V : d(v, z) < d(x, z)\}$.*

### 3.2.2 Distribution and Hitting Probability

Consider an underlying graph $H = (V, E)$, which may be directed but must be connected in the sense that it contains a direction respecting path from any vertex to any other. Let $d(x, y)$ be the distance function implied by $H$, and let a random graph $G$ be constructed by augmenting

$H$ with one random directed edge starting at each vertex. The edges added by the augmentation will be denoted as $\gamma : V \mapsto V$. We call $\gamma$ a *shortcut configuration*, and let $\Gamma = V \mapsto V$ be the set of all such functions. The general probability space over which we will work is $\Gamma \times V \times V$, where the two copies of $V$ represent the possible starts and destinations of walks. Let $\mathbf{P}$ be a probability measure on this set where the start and destination are chosen uniformly and independently of each other and the configuration is chosen by some *shortcut distribution* $\ell(\gamma)$ which in the independent selection case may be factored into the form $\prod_{x \in V} \ell(x, \gamma(x))$.

On this space, we define $X_Z^Y(t)$ for $t = 0, 1, \ldots$ as a greedy walk in the graph from a uniformly chosen starting point $Y = X_Z^Y(0)$ with a uniformly chosen destination $Z$. To make the greedy walk well defined, we dictate that ties are broken randomly (that is, if the $m$ closest neighbors to the destination are equally far from it, one is selected uniformly at random.) Below, we will in particular be interested in the hitting probability of greedy walks with specific destinations. We define this formally as

$$h(x, z) = \mathbf{P}(X_Z^Y(t) = x \text{ for some } t | Z = z) \tag{3.2}$$

If $H$ is a translation invariant graph, then $h(x, z) = h(x - z, 0)$ for some distinguished vertex 0. Thus we will, without further loss of generality, consider the hitting probability as a function of one variable and write $h(x) = h(x, 0)$. Further, we will restrict our analysis to cases where $\ell(x, y)$ and $h(x, y)$ are functions of $d(x, y)$ only (we call this *distance invariance*).

Our results concern relating $h(x)$ to the occurrence of shortcuts between vertices. 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

$$T = \sum_{x \in V} \chi_{\{X_0^Y(t) = x \text{ for some } t\}}$$

whence it follows that

$$\mathbf{E}[T] = \sum_{x \in V} h(x). \tag{3.3}$$

We will denote the expected greedy walk length $\tau = \mathbf{E}[T]$.

## 3.3 Rewiring by Destination Sampling

Before proceeding to analyze our main model, we present the rewiring algorithm which motivates it. Running the algorithm modifies, in each step, the destinations of some shortcut edges. It is a steady-state algorithm in the sense that the number of edges never changes: 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 graphs arise, this is similar to the celebrated preferential attachment model for power law graphs of Barabási and Albert. However, it is a not a growth model for the graph since the number of vertices and edges never changes, and is thus more similar to the variant of preferential attachment discussed in [25].

The proposed algorithm, which we call *destination sampling*, is as follows:

**Algorithm 3.3.1.** *For a given graph $H = (V, E)$, let $\gamma_s$ be a shortcut configuration at time $s$. From each vertex there is exactly one shortcut. Let $0 < p < 1$. Then $\gamma_{s+1}$ 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 $z_s$ using $H$ and the shortcuts of $\gamma_s$. 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 $z_{s+1}$ (that is let $\gamma_{s+1}(x_i) = z_{s+1}$).*

After a walk is made, $\gamma_{s+1}$ is the same as $\gamma_s$, except that the shortcut from each vertex 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. We strongly believe that updating the shortcuts using this algorithm eventually results in a shortcut graph with greedy path-lengths of $O(\log^2 n)$. Though one can relate the stationary regime of this algorithm

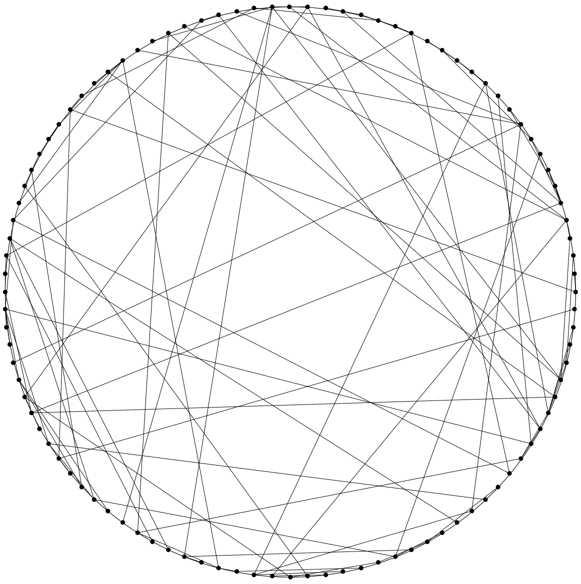


Figure 3.1: A shortcut graph generated by our algorithm ($n = 100$).

to the balanced distributions (see below), a rigorous bound has not been proved.

The value of $p$ is a parameter in the algorithm. It serves to disassociate the shortcut of a vertex with those of its neighbors. For this purpose, the lower the value of $p > 0$ the better, but very small values of $p$ will also lead to slower sampling.

### 3.3.1 Markov Chain View

Each application of Algorithm 3.3.1 defines the transition of a Markov chain on the set of shortcut configurations, $\Gamma$. The Markov chain in question is defined on a finite (if large) state space. If it is irreducible and aperiodic, it thus converges to a unique stationary distribution.

**Proposition 3.3.2.** *The Markov chain* $(\gamma_s)_{s\geq 0}$ *is irreducible and aperiodic.*

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

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 $y$ to point at a given target $z$ without changing the rest of the graph. But the probability of this happening in a single iteration is at least

$$\mathbf{P}(Y = y, Z = z, \text{ and only } y \text{ re-wired}) \geq \frac{1}{n}\frac{1}{n}p(1-p)^{n-2} > 0.$$

□

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.

**Proposition 3.3.3.** *Under the unique stationary distribution of the Markov chain $(\gamma_s)_{s\geq 0}$*

$$\ell(x, z) = \frac{h(x, z)}{\sum_{\xi\neq 0} h(\xi)}.$$

*Proof.* As selected by the algorithm, 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, this can be seen as a statement relating $\ell$ to the hitting probability, i.e.

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

The first multiple in the numerator is the hitting probability $h(x, z)$. The formula then follows from the uniform distribution of $Z$ and translation invariance. □

## 3.4 Balanced Shortcut Distributions

We use Proposition 3.3.3 as the starting point of our analysis, defining the class of all distributions having the same marginal property as follows.

**Definition 3.4.1.** *If a graph $H$ with distance function $d(x, y)$ is randomly augmented such that*

$$\ell(x, z) = \frac{h(x, z)}{\sum_{\xi \neq 0} h(\xi)} \tag{3.4}$$

*where $h$ is given by (3.2), then the joint distribution of shortcuts is said to be* balanced*.*

We will show for several classes of graphs that this relationship leads to navigability, allowing for a characterization other than that of equation (3.1). Besides the relationship with Algorithm 3.3.1, balance is in some ways a natural requirement. The left hand side describes the distribution of the destinations of walks that hit the point $x$, so our results simply say that a good choice of shortcuts is one that matches this.

**Theorem 3.4.2.** *For a translation invariant graph $H$, there exists a balanced distribution which selects shortcuts independently at each vertex.*

*Proof.* Like before, we let $\ell(x, y)$ be the marginal probability that $x$ has a shortcut to $y$. The joint distribution is simply the product over all vertices.

For a single walk toward a given $z$, we may view $X_z^Y(t)$ as a 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 calculations below 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 closer to 0 than $y$ because it is the destination of the shortcut from $y$, or we step to one of $y$'s neighbors in $H$ because $y$'s shortcut leads to somewhere from which it is further to 0 than $y$.

Let $N(x)$ be the set of neighbors of $x$ in $H$, and $L(x) = \{\xi \in V : d(\xi, 0) \geq d(x, 0), \xi \neq x\}$ be the set of vertices at least as far as $x$ from 0.

Also, let $P(x) = \{\xi \in N(x) : d(\xi, 0) > d(x, 0), (\xi, x)$ edge in H$\}$ (the set of "parent" vertices that can greedy route to $x$ in $H$) and $C(x) = \{\xi \in N(x) : d(\xi, 0) = d(\xi, 0) > d(x, 0), (x, \xi)$ edge in H$\}$ (the set of "child" vertices that $x$ can route to). Then the transition kernel of the process described is

$$P(y,x) = \begin{cases} 0 & \text{if } d(x,0) \geq d(y,0), x \notin C(y) \\ \ell(y,x) + \frac{1}{|C(y)|} \sum_{\xi \in L(y)} \ell(\xi) & \text{if } x \in C(y) \\ \ell(y,x) & \text{if } d(x,0) < d(y,0) - 1 \end{cases}$$

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

We can thus express the hitting probability for any $x \neq 0$ for a greedy walk as

$$\begin{aligned} h(x) &= \sum_{\xi \in V : \xi \neq x} h(\xi)\ell(\xi, x) + \frac{1}{n-1} \\ &= \sum_{\xi : d(\xi,0) > d(x,0)} h(\xi)\ell(\xi, x) + \sum_{\xi \in P(x)} h(\xi) \sum_{\eta \in L(\xi)} \frac{\ell(\eta, \xi)}{|C(\xi)|} \\ &\quad + \frac{1}{n-1}. \end{aligned} \tag{3.5}$$

The first two terms in (3.5) represent the probability that we enter $x$ through either a shortcut or from a parent vertex respectively, and the last term is the probability that the walk starts at $x$.

Note that, for any $x$, equation (3.5) gives a recursive definition of $h(x)$ in terms of the distribution $\ell$. Fix such a distribution $\ell'$. From this we can thus calculate the hitting probabilities $h'(x)$, and define

$$\ell''(x) = \frac{h'(x)}{\sum_{x \in V \setminus \{0\}} h'(x)}.$$

The mapping $\ell' \mapsto \ell''$ is continuous since $\sum_{x \in V} h'(x) > 1$ and maps the simplex of $(n-1)$ valued distributions into itself. Since the simplex is convex, Brouwer's fix-point theorem gives the existence of at least one fix-point $\ell^*$, which is a balanced distribution. □

## 3.5 The Directed Cycle

We let $H$ be the directed cycle on $n$ vertices, which will be numbered 0 through $n-1$ such that the edges are directed downward (modulo $n$). The implied distance function (which is not symmetric) is

$$d(x,y) = \begin{cases} x - y & \text{if } y \leq x \\ n - y + x & \text{otherwise.} \end{cases}$$

This environment is perhaps the most natural one for greedy routing, and has previously been the subject of a thorough analysis by [6]. There exists exactly one point at each distance from 0, and greedy routing means selecting the shortcut if and only if its destination lies between 0 and the current position. Equation (3.5) here simplifies to:

$$h(x) = \sum_{\xi=x+1}^{n-1} h(\xi)\ell(\xi - x) + h(x+1) \sum_{\xi=x+2}^{n-1} \ell(\xi) + \frac{1}{n-1}.$$

To prove our result in this environment, we will need the following lemma:

**Lemma 3.5.1.** *If the shortcut configuration is chosen according to a distance-invariant joint distribution, then $h(x)$ is non-increasing in $x$.*

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

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

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 $\gamma$, such that $(\gamma, y) \in I$. This means that there is an integer $m$ and a path $x_0, x_1, \ldots, x_m$ 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 $i = 0, 1, \ldots, m$. 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 $i = 0, 1, \ldots, m$. 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 $n - 2$ rather than $n - 1$ in the first line).

It now follows directly that

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

□

We can now show that greedy routing here has takes a similar number of steps to the critical case in Kleinberg's model.

**Theorem 3.5.2.** *For every $n = 2^k$ with $k \geq 3$, 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 introduced by Kleinberg for augmentations described by Equation (3.1) 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 $\{1, 2, \ldots, n - 1\}$ 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 the same number of steps in each phase. Thus, the first phase is the interval $F_1 = \{1, \dots, r_1\}$ 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 shortest interval $\{r_1 + 1, \dots, r_2\}$ such 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 the remaining interval $\{r_m + 1, \dots, n-1\}$ which could be empty. By construction $\ell(F_R) < 1/k$ and the total number of phases, including $F_R$, is at most $k$.

Before proceeding, we need to bound by how much $\ell$ of the different phases can deviate from one another since this will also tell us by how much the expected number of steps in each phase can differ. From (3.4) and the assumed lower bound of $\tau$, it follows that

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

for all $x$. This implies that

$$\frac{1}{k} \leq \ell(F_i) \leq \frac{1}{k} + \frac{1}{2k^2}$$

for all $i \in \{1, \dots, m\}$, and thus

$$\ell(F_i) \leq \left(1 + \frac{1}{2k}\right) \ell(F_j) \tag{3.6}$$

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

Consider now $F_m = \{r_{m-1} + 1, r_{m-1} + 2 \dots, r_m\}$ and let $L = \{0, 1, \dots, r_{m-1}\}$. Our goal is to show that, from any point in $F_m$, there is a considerable probability of having a shortcut to $L$. We know that $r_m \leq n$. 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

$$\{r_m - F_m\} \subset L$$

Thus, if $r_m$ has a shortcut with destination in $\{r_m - F_m\}$, any walk which hits $r_m$ will leave $F_m$ in the next step (see Figure 3.2). We thus know that

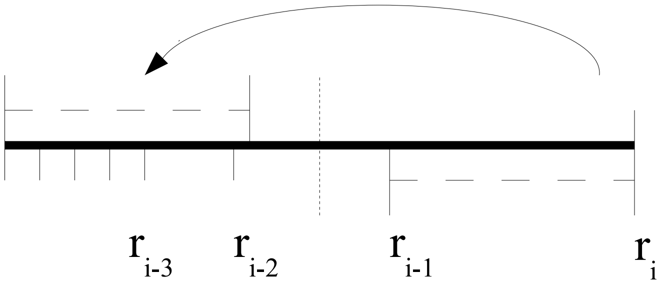


Figure 3.2: Illustration of the proof of Theorem 3.5.2. If a phase covers less then half of the "remaining ground", then any shortcut of the same distances from $r_i$ as the 0 is from the points in the phase takes us out of the phase.

$$\ell(r_m, L) \geq \ell(r_m, \{r_m - F_m\}) = \ell(F_m) \geq 1/k.$$

Lemma 3.5.1 tells us that the probability of having a shortcut to $L$ cannot decrease for points less than $r_m$, so for each vertex that the walk hits within $F_m$, there is an independent probability of at least $1/k$ of leaving $F_m$ in the next step. This means that the expected number of steps the walk can take in $F_m$ is at most $k$.

The expected number of steps in a phase is $h(F_i) = \tau\ell(F_i)$ so, by (3.6), it then holds that

$$h(F_i) \leq (1 + 1/2k)h(F_m) \leq k + 1/2 \tag{3.7}$$

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 in particular

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

This means that $F_1$ contains at most 4 points, so $h(F_1) \leq 4$ and thus, again by (3.6), $h(F_i) \leq 5$ for all $i$. For $k \geq 3$ this contradicts the original assumption. This completes the proof. □

Theorem 3.5.2 gives us an alternative distributional criterion for attaining $O(\log^2 n)$ expected greedy path-lengths. Since Kleinberg showed that this cannot hold for many distributions, the balanced distributions must be "close" to the critical, harmonic decay. More specifically, drawing on the proofs that navigability is not possible for most cases graphs, we can see that there cannot exist $\delta > 0, \epsilon > 0$ and $N \in \mathbb{N}$ such that $\ell(\{1, 2, \ldots, n^\delta\}) \leq n^{-\epsilon}$ for the cycles of size $n \geq N$. This would be the case if the tails of the distributions dominated a power-law ($x^{-\alpha}$) decay with exponent $\alpha < 1$. Similarly, there cannot exist (possibly different) $\delta > 0, \epsilon > 0$ and $N \in \mathbb{N}$ such that

$\ell(\{n^\delta, n^\delta + 1, \ldots, n-1\}) \leq n^{-\epsilon}$ for the cycles of size $n \geq N$, as would be the case if the tails were dominated by a power-law with exponent $\alpha > 1$.

## 3.6 Delaunay Graphs

The small-world theory is not necessarily limited to situations where vertices are placed in a fixed grid. In this section, we will let the vertices be points of a spacial Poisson process, and the distance function be the Euclidean metric. For simplicity, we will relax our requirements a little and let the graphs have degrees bounded in expectation, rather than uniformly.

Let $S^d$ be the $d$-dimensional surface of a $d+1$ ball with radius such that the volume/area of $S^d$ is 1. We let $V = \{x_i\}$ be the $N$ points of a homogeneous Poisson process with intensity $\lambda = n^d$ in this space. From this Poisson process we may construct the Voronoi tessellation, that is the collection of cells $C(x_i)$ where

$$C(x_i) = \{y \in \mathbb{R}^d_s : |y - x_i| = \min_{z \in V} |z - x_i|\}.$$

$C(x_i)$ is that part of the space which is as close to $x_i$ as to any other point. The Voronoi cells are closed convex polyhedrons that border other cells along each side, thus overlapping on sets of Lebesgue measure zero.

The tessellation induces a graph $G$ with vertices $V$ (known as the Delaunay graph) as follows. Let $G = (V, E)$, where $(x, y) \in E$ if and only if $C(x)$ and $C(y)$ intersect in an infinite number of points (this is a.s. equivalent to intersecting in at least one point). Intuitively, this is the graph that connects a vertex $x$ to all its neighbors in the tessellation. This Delaunay graph is a natural base graph for greedy routing among the points.

**Lemma 3.6.1.** *Let $\{x_i\}$ be any point-set in $S^d$, and $G$ its Delaunay graph. Then the Euclidean metric $d(x, y) = |x - y|$ is adapted for routing in $G$.*

*Proof.* We must prove that for all $x \neq z \in V$, there exists $y \in V$ (which may be $z$) such that $(x, y) \in E$ and $|x - z| > |y - z|$. Consider the line $xz$. Let $w$ be the first point we encounter as we move from $x$ along $xz$, satisfying $w \in C(y)$ for some $y \neq x$ ($w$ is well-defined since the cells are compact).

It is clear that $w \in C(y)$ for some $y$ such that $x$ and $y$ are connected in the Delaunay graph ($C(x)$ must border at least one cell that it meets at $w$). Clearly, $|y - w| = |y - w|$ since $w$ is in both cells. Thus

$$\begin{aligned} |y - z| &< |y - w| + |w - z| \\ &= |x - w| + |w - z| = |x - z| \end{aligned}$$

where the strict inequality follows from the fact that $w$ is not on the line $yz$. □

Given this graph, we consider augmentations that allow for fast routing. A direct approach would be to connect a given vertex to any other with a probability depending on the distance between them, but this leads to complications regarding dependencies between the progress made at each step (though not insurmountable ones, see [21] for such an approach in a similar environment).

Instead, we augment the graph as follows. For each vertex $x \in V$, let $\{n_i(x)\}_{i=1}^{N(x)}$ be the points of a non-homogeneous Poisson process given by the measure $\mu_x(A) = \ell_x(A \backslash C(x))$ for some *shortcut measure* $\ell$ on the Borel sets of $S^d$, and $\ell_x(A) = \ell(A - x)$.

We then augment $G$ by adding a directed edge from $x$ to $y$ if $n_i(x) \in C(y)$ for any $i = 1, \ldots, N$.

**Lemma 3.6.2.** *If $x, z \in V$ and $|z - n_i(x)| \leq |z - x|/4$ for some $i = 1, 2, \ldots, N$. Then $x$ has a shortcut $y \in V$ (which may be $z$) such that $|z - y| \leq |z - x|/2$.*

*Proof.* Let $w$ be such an $n_i(x)$. With probability one it is in exactly one cell $C(y)$. If $y = z$ then $x$ has a shortcut to $z$, otherwise $|w - y| < |w - z|$. In the latter case,

$$|z - y| \leq |z - w| + |w - z| < 2|w - z| \leq |x - z|/2.$$

□

### 3.6.1 Kleinberg Augmentation

To motivate the model, we first show that augmentation along the lines of Kleinberg's model allows for an $O(\log^2(n))$ bound on the routing time. That is, as in Equation (3.1), we let the augmentation be given by the measure

$$\ell(A) = \int_A \frac{dr}{\log n \mathrm{Vol}(r)} \tag{3.8}$$

where $\mathrm{Vol}(r)$ is the volume of a ball of radius $r$ in $S^d$. The measure is defined on sets $A \in S^d \backslash \{0\}$.

Before proving a lower bound on the expected routing type, we need to ensure that we are not adding an unbounded number of edges.

**Lemma 3.6.3.** *The expected number of shortcuts added to each vertex under augmentation with intensity (3.8) is bounded by a constant.*

*Proof.* First note that $\mathbf{E}[\#\text{shortcuts added to } x] \leq \mathbf{E}[N(x)]$. Now, let $R(x) = \inf\{|y-x| : y \in V, y \neq x\}$. If $R(x) = \delta$, then all points within distance $\delta/2$ of $x$ are in $C(x)$. Thus

$$\begin{aligned}
\mathbf{E}[N(x) \,|\, R(x) = \delta] &\leq \frac{1}{\log n} \int_{S^d \backslash B_{\delta/2}(x)} \frac{1}{\mathrm{Vol}(x-y)} dy \\
&\leq \frac{1}{\log n} \int_{\delta/2}^{1} \frac{1}{r} dr = \frac{\log(2/\delta)}{\log n}
\end{aligned}$$

Hence, and since $\mathbf{E}[N(x) \,|\, R(x) = \delta]$ is decreasing in $\delta$,

$$\begin{aligned}
\mathbf{E}[N(x)] &= \int_0^1 E[N(x) \,|\, R(x) = \delta] f_{R(x)}(\delta) d\delta \\
&\leq \mathbf{E}[N(x) \,|\, R(x) = 1/n] \mathbf{P}(R(x) \geq 1/n) \\
&\qquad + \int_0^{1/n} \frac{\log(2/\delta)}{\log n} n^d S(\delta) e^{-n^d \mathrm{Vol}(\delta)} d\delta \\
&\leq 2 + \frac{n^d S(1/n)}{\log n} \int_0^{1/n} \log(2/\delta) d\delta \\
&\leq c,
\end{aligned}$$

where $S(\delta)$ is the area of a sphere of radius $\delta$, and $c$ is a constant independent of $n$. □

The proof of the following theorem uses the by-now standard argument from [39].

**Theorem 3.6.4.** *For every $n$ sufficiently large, the shortcut graph created by augmenting the Poisson-Delaunay graph with intensity (3.8) has an expected greedy routing time of $O(\log^2 n)$.*

*Proof.* Let the route currently be at the vertex $x$, such that $|x - z| = d > 1/n$. Let $B$ be the event that $|n_i(x) - z| \leq d/4$ for some $i$. Then

$$\mathbf{P}(B) \geq \frac{\mathrm{Vol}(d/4)}{\mathrm{Vol}(3d/4)\log n} = \frac{c}{\log n}.$$

By Lemma 3.6.2, if such a $n_i(x)$ exists, then $x$ has a neighbor within distance $d/2$ of $z$, and greedy routing at least halves the distance to $z$ in the next step. If $B$ fails to occur, then we know by Lemma 3.6.1 that greedy routing can still progress to a point closer to the destination, and whether or not $B$ occurs is independent of previous steps. Thus the expected number of steps until the distance to the target is halved is $O(\log n)$, which together with Lemma 3.2.2 proves the result. □

### 3.6.2 Balanced Augmentation

In order to derive a result similar to Theorem 3.5.2 for the Delaunay setting, we will need to re-define the "balanced distribution" somewhat. In particular, we need to marginalize over the positions of the Poisson points.

Let the hitting measure of a $A \subset S^d \backslash \{0\}$ be defined by:

$$h_z(A) = \mathbf{E}[\text{number of } t \text{ s.t. } X_Z(t) \in A \,|\, Z = z]$$

where $X_z(t)$ is the greedy routing process as above, and the existence of a point at $z$ is included in the conditioning. Note that, by the translation invariance of the construction, $h_z(A) = h_0(A - z)$.

We call a distribution *Poisson-balanced* if

$$\ell(A) = \frac{h_0(A)}{\tau} \tag{3.9}$$

where $\tau = \mathbf{E}[\text{length of a greedy walk}] = h_z(S^d \backslash \{0\})$.

**Lemma 3.6.5.** *There exists a Poisson-balanced distribution.*

*Proof.* The proof is similar to the discrete case. A given shortcut measure $\ell'$ induces a hitting measure $h_0'(A)$, which in turn gives rise to a measure $\ell''$ via (3.9). If we let $L$ be the space of measures of total probability one on $S^d \backslash \{0\}$ equipped with the total variation metric

$$d_{\mathrm{TV}}(\mu, \nu) = \sup_{A \in \mathcal{B}(S^d\{0\})} |\mu(A) - \nu(A)|,$$

then the mapping $\ell' \mapsto \ell''$ is a mapping from $L$ to itself. $L$ is convex and compact, so it suffices to show that the mapping is continuous for us to apply Brouwer's fix-point theorem.

Since we know that $\tau > 1$, the second step of the mapping is certainly continuous. The first is also, since the hitting probability depends only on a finite number of random variables with distribution depending on $\ell'$. Formally:

Take $\epsilon > 0$ and any $m = n^d$. Let $\ell_1$ and $\ell_2$ be two shortcut measures. Without loss of generality, we assume that $\ell_2 \geq \ell_1$, and we let $d_{\mathrm{TV}}(\ell_1, \ell_2) \leq \epsilon'$ where

$$\epsilon' \leq \frac{\epsilon}{3m \max((e-1)n, \log(3m/\epsilon))}.$$

We couple the routing processes $X_0^1$ and $X_0^2$ by letting them use the same set of Poisson process distributed vertices $V$, and the same starting point $z$. At each $x \in V$, we construct shortcuts $n_i(x)$ according to $\ell_1$ which both processes may use, and then add an additional set of shortcuts $\{n_i^2(x)\}$ according to $\ell_2 - \ell_1$, which only $X_0^2$ may use.

It follows that for any $x$, the cardinality of $\{n_i^2(x)\}$, $N(x)$, is dominated by a $\mathrm{Poi}(\epsilon')$ random variable, so

$$\mathbf{P}(N(x) = 0) \leq 1 - e^{-\epsilon'} \leq \epsilon'$$

Let $B$ be the event that a given vertex $x$ in $V$ has $N(x) > 0$. Then

$$\begin{aligned}
\mathbf{P}(B) &\leq \mathbf{P}(B \,|\, |V| \leq (e-1)m + q) + \mathbf{P}(|V| \leq (e-1)m + q) \\
&= ((e-1)m + q)\epsilon' + e^{-q} \\
&\leq \frac{\epsilon}{m},
\end{aligned}$$

where the last inequality follows from setting $q = \log(3m/\epsilon)$.

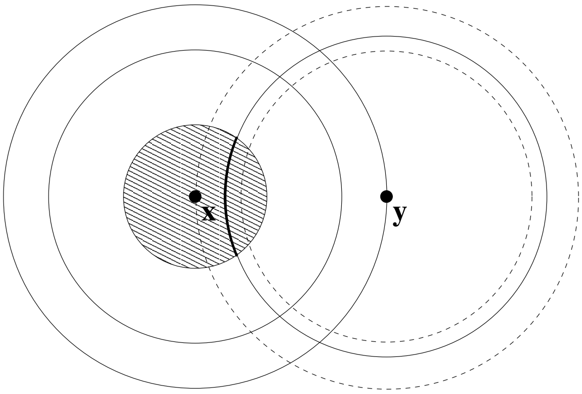


Figure 3.3: The circle around $y$ intersects the ball around $x$ in at least 1/8 of its points.

Now let $H_1(A)$ and $H_2(A)$ be the number of points reached in a subset $A \subset S^d \backslash \{0\}$ by the respective processes. If the $h_0^1$ and $h_0^2$ are the respective hitting probabilities, then

$$\begin{aligned} |h_0^1(A) - h_0^2(A)| &= \mathbf{E}[|H_1(A) - H_2(A)|] \\ &= \mathbf{E}[|V|]\mathbf{P}(H_1(A) \neq H_2(A)) \leq \epsilon \end{aligned}$$

since $H_1(A) = H_2(A)$ if no vertex has different shortcuts in the two cases. This completes the proof. □

In order to bound the routing time in this case, we will need the following geometrical fact.

**Lemma 3.6.6.** *There exists $q \in (0,1)$ such that, if $x$ and $y$ are points in a $S^d$, satisfying $(3/4)\delta < |x-y| \leq \delta$, and $(3/4)\delta < r \leq \delta$, then the portion of the sphere $S_r(y)$ which lies inside $B_{(3/8)d}(x)$ is at least $q$. The constant $q$ depends on $d$ but not on $\delta$ and $r$.*

This follows directly from the fact that the statement is independent of scale. In one dimension $q = 1/2$ trivially, and in two it can easily be seen that it is at least 1/8, see Figure 3.3.

**Theorem 3.6.7.** *For every sufficiently large $k$ and $n = \left(\frac{4}{3}\right)^k$, the shortcut graph created by augmenting the Poisson-Delaunay graph with* *a Poisson-balanced distribution has an expected greedy routing time* $\tau \leq \frac{k^2}{q}$.

*Proof.* We let $X_0^Y$ be the routing process for zero, and define $h_0$ on $S^d \backslash \{0\}$ as above. We then divide $S^d \backslash \{0\}$ into $k$ phases of the form $F_i = \{x \in S^d : r_{i-1} < |x| \leq r_i\}$, where $r_0 = 0$ and each subsequent $r_i$ is defined so that:

$$h_0(F_i) = \frac{\tau}{k}.$$

For any phase $F_i$, assume that $r_{i-1} \geq \frac{3}{4} r_i$. Let $x$ be a vertex in $F_i$. A portion $q$ of the area of each spherical "level" in $x + F_i$ lies in $L_i = B_0((3/8)r_i)$ by Lemma 3.6.6. By rotational invariance it follows that $\ell_x(L_i) = q\ell_x(x + F_i) = q\ell(F_i)$, so if $B$ is the event $x$ has a shortcut destination $n_i(x)$ closer than $r_{i-1}/2$, then

$$\mathbf{P}(B) = \ell_x(B_0((3/8)r_i)) \geq q\frac{h_0(F_i)}{\tau} = \frac{q}{k}.$$

By Lemma 3.6.2, if such a $n_i(x)$ exists, then $x$ has a neighbor within distance $d/2$ of $z$, and greedy routing at least halves the distance to $z$ in the next step. If $B$ fails to occur, then we know by Lemma 3.6.1 that greedy routing will progress to vertex closer to the target. The event $B$ is independent of previous steps. Thus $h_0(F_i) \leq \frac{k}{q}$, whence $\tau \leq \frac{k^2}{q}$.

If, on other hand $r_{i-1} \leq \frac{3}{4} r_i$ for all $i$, then

$$r_1 \leq \frac{3}{4}^{k-1} = \frac{4}{3n}.$$

Let $N$ be the number of vertices in $F_1$. By Observation 3.2.2

$$h_0(F_1) \leq \mathbf{E}[N] = \frac{\mathrm{Vol}(r_1)}{n^d} = c$$

It follows that $\tau \leq ck$, so the result holds when $k > cq$. □

## 3.7 The Rewiring Algorithm Revisited

Proposition 3.3.3 shows that, under the stationary distribution of the destination sampling algorithm introduced above, the marginal shortcut distribution at each point is balanced, and it is thus tempting to apply

Theorem 3.5.2 to bound the greedy path-length. However, that theorem assumed that the shortcuts had been chosen independently at each vertex, which is not the case under Algorithm 3.3.1 which originally motivated the work. Showing that these dependencies do not negatively affect routing is an open problem, which we discuss in general terms in this section.

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 $x$ sees. In the directed cycle, if $x + 1$ has a shortcut to $x - 10$, that will make it less likely for $x$ to see walks for places "beyond" $x - 10$, since many such walks will have followed the shortcut at $x + 1$, and thus skipped over $x$.

The first source of 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, 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 that 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 values of $p$. Although it is more complicated, it is easier 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 routing. 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 3.5.2 we use independence only to show that if the probability of having a shortcut out of a phase at the very furthest point is $\rho$, then the expected number of steps in the phase is bounded by $1/\rho$. There is little reason to believe this wouldn't hold under the modified algorithm, since if the link from the furthest point doesn't take us out of the phase, then 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 those we see in the future. If, on the other hand, it overshoots, then by the argument above it should make it more likely that the succeeding ones do not do so, giving a us a better probability of leaving the phase than in the independent case.

Formalizing the requirements on the dependence, and proving that our stationary distribution indeed has the necessary properties, is the main open problem which we have yet to resolve.

### 3.7.1 Computer Simulation

Simulations indicate that the algorithm gives results which scale as desired in the number of greedy steps, and that the distribution approximates $1/(h_{n,d}d(x,y))$ for the one dimensional grid.

The results in the directed one-dimensional grid can be seen in Figure 3.4. To get these results, the graph is started with no shortcuts, and then the algorithm is run $10n$ times to initialize the references. The value $p = 0.1$ 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 to lower the variance of the observed value.

The square root of the mean greedy distance increases linearly as the graph 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 graph size is found to increase the square root of the greedy distance by about 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 Kleinberg's model we can use (3.5) to calculate numerically exact values for $\tau$, allowing us to confirm this figure.)

In Figure 3.6 the marginal distribution of shortcut lengths is plotted. It is roughly harmonic in shape, except that destination sampling creates less links of length close to the size of the graph. This may be part of

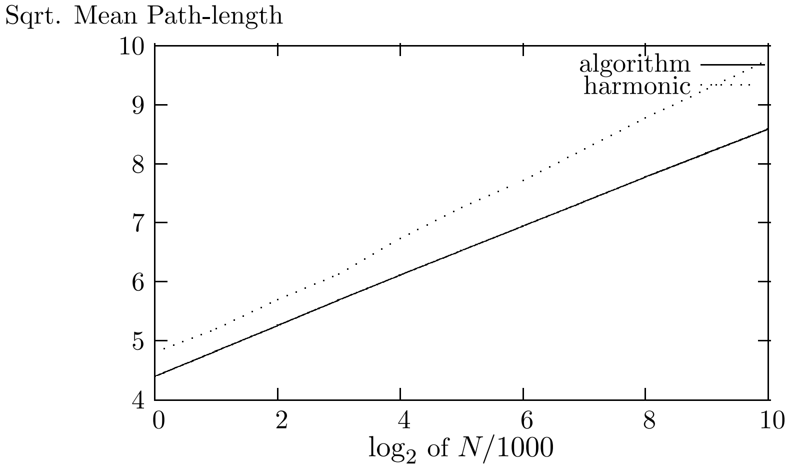


Figure 3.4: The expected greedy walk time of the selection algorithm, compared to selection according to harmonic distances, in a cycle.

the reason why it is able to outperform Kleinberg's model: while the latter is asymptotically correct, our algorithm takes into account finite size effects. (This reasoning is similar to that of the authors of [16]. 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 3.5 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.

## 3.8 Conclusion

The study of navigable graphs 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 graphs to be navigable, and we have presented an algorithm which may explain how navigable graphs arise naturally. The algorithm's simplicity also

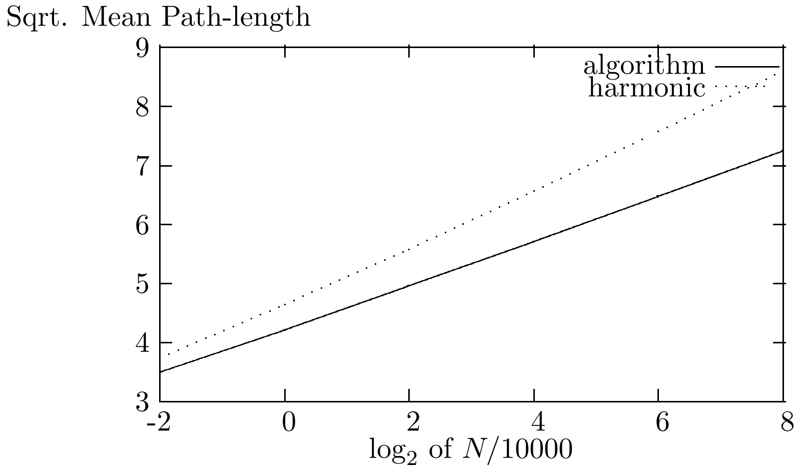


Figure 3.5: The expected greedy walk time of the selection algorithm, compared to selection according to harmonic distances, in a two dimensional base grid.

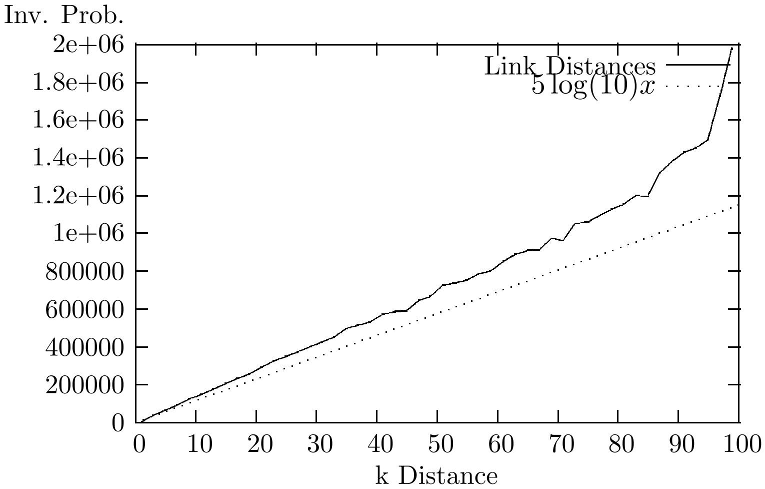


Figure 3.6: The inverse of the distribution of shortcut distances, with $n = 100000$, $p = 0.1$. The straight line is the inverse of the harmonic distribution.

means that it can be useful in practice for generating graphs that can easily be searched, an important property for many structures on the Internet.

While many questions about these graphs in general, and our results in particular, remain unanswered, the prospects for 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 graphs of all kinds.

## Acknowledgments

Thanks to Ian Clarke, who originally proposed the link updating scheme discussion with me. Thanks to my advisers Olle Häggström and Devdatt Dubhashi for their help, as well as Jon Kleinberg for listening to and reflecting on my ideas.

# Chapter 4

# Double Clustering

## 4.1 Introduction

Motivated by the small-world experiments of Stanley Milgram [48], and the models for social networks inspired by them [62], Jon Kleinberg introduced the question of whether graphs can be searched (or navigated) in a decentralized manner [39]. In particular, he showed that when a grid structure is augmented by random edges, whether it is possible to use those edges to efficiently route queries from one point in the grid to another depends on their distribution. In particular, if each vertex $x$ in a $d$-dimensional grid is given one additional "long range" link to some vertex, beyond those to its nearest neighbors, then when the probability of $y$ being the selected is proportional to $|x - y|^{-d}$, any greedy walk on the resulting graph is expected to complete in a number of steps polylogarithmic to the graph size. If the probability of $y$ being selected is any other exponent of the distance (in particular 0, meaning the long-range link is uniformly selected) then any form of routing which uses only information about the points seen thus far will require a number of steps which is a fractional power of the graph size.

A natural question following from Kleinberg's results is to ask if there is any dynamic which might cause the frequency of edges in naturally occurring graphs to have the sought relationship with their length. Several empirical studies of social network data following Kleinberg have observed just such a relationship [2] [44], making it plausible that such a dynamic may exist, but it has not been identified.

In this paper we observe that the desired edge distribution arises naturally in another probabilistic model, that of best-yet sampling from a population, and use this to show how navigable networks may arise when vertices belong to two independent spaces and tend to cluster in both (in social network terms, these may be identified with the physical world and a metaphorical space of "interests" – people tend to be befriend those who are close in either sense.) The resulting spatial random graph, which we dub the double clustering graph, turns out to be navigable with respect to both spaces.

### 4.1.1 Previous Work

The original questions about navigability in a geographical setting were posed and answered by Kleinberg in [39] and [38]. Later, Kleinberg [40] and Watts et al. [61] independently proposed similar models based on the categorization of ideas or characteristics. The latter paper includes the idea that vertices may be similar in several independent spaces, but does not discuss how this might lead to the desired edge distribution. Fraigniaud [28] went further and discussed augmentation in more general settings based on tree-decompositions of general graphs.

Some conceptually different work has been done previously to try to explain the emergence of Kleinberg type edge frequencies. In particular, [16] [56] and [55] propose graph rewiring processes which seem to create navigable networks in their stationary state. These may help explain how such networks arise under some circumstances, but are not always an easy fit with observed reality, and have so far eluded complete analysis. [23] shows a form of navigable augmentation that depends on little knowledge of the base graph, but this algorithm is complicated and does not give an intuitive reason why the desired edge distribution should arise.

### 4.1.2 Contribution

We characterize our contribution as follows:

- We introduce the "double clustering" graph construction. This is a simple rule for constructing a graph between a set of vertices with positions in two different spaces, so that they tend to connect to those nearest in both. Double clustering can be seen as a spatial or combinatorial construction depending on whether the points are originally placed or metric spaces in graphs.

- We show analytically that in several cases double clustering leads to navigable graphs.
- We hypothesize that this holds for a much larger class of such graphs, something we illustrate with simulation of several relevant sub-models.

## 4.2 Navigable Graphs

### 4.2.1 Decentralized Routing

Let $G = (V, E)$ be a connected finite graph of high (some power of $|V|$) diameter, and let the random graph $G'$ be created by addition (augmentation) of random edges to $G$. It is well known, see for instance [11], that the diameter shrinks quickly to a logarithm of $|V|$ when random edges are added between the vertices. Navigability concerns not a small diameter, however, but rather a stronger property: the possibility of finding a short path between two vertices in $G'$ using only local knowledge at each vertex visited. By local knowledge, one means that each vertex knows $G$, but does not know which random edges have been added to any vertex until it is visited. The exact limits of such *decentralized routing algorithms* have been discussed elsewhere [39] [6], but since we are interested only in upper bounds, we will define only the subset of such algorithms of interest to us. When routing for some vertex $z$, in each step we will select as the next vertex in the path a $G'$-neighbor of the current vertex, $x$. This choice will be made entirely as a function of each neighbors $G$-distance to $z$, and nothing else. All such algorithms are decentralized by Kleinberg's definition.

The most direct decentralized algorithm, and the most important, is *greedy routing*. In greedy routing, the next vertex chosen is that neighbor which is closest to $z$ in $G$ (with some tie-breaking rule applied). Note that both the original and augmented edges can be used, but because the choice is only optimal with respect to $G$, the path discovered by greedy routing will seldom be a minimal path in $G'$. In one case below we will modify the routing to divert from a greedy choice slightly for technical reasons, but the principle is still the same.

We start with $G$ as a $d$-dimensional $n$-grid (that is $V = \{1, 2, \ldots, n\}^d$ and there are edges between adjacent vertices) and independently add a single directed edge from each vertex to a random destination. The long-range connection is added such that for $x, y \in V$, and some $\alpha \geq 0$

$$\mathbf{P}(x \rightsquigarrow y) = \frac{1}{h_{\alpha,n}|x-y|^{\alpha}} \tag{4.1}$$

where $x \rightsquigarrow y$ is the event that $x$ is augmented with an edge to $y$ and $|x - y|$ denotes $L^1$ distance in $\mathbb{Z}^d$. $h_{\alpha,n}$ is a normalizing constant.

The by now well known result of Kleinberg is that when $\alpha = d$, greedy routing between any two points in $V$ takes $O(\log^2 n)$ steps in expectation, while for any other value of $\alpha$ decentralized algorithm creates routes of expected length at least $\Omega(n^s)$ steps for some $s > 0$ (where $s$ depends on the dimension but not the algorithm chosen).

### 4.2.2 Doubling Size and More General Augmentation

It should be noted that if the graph is a $d$-dimensional grid as above, and for $x \in V$ $B_r(x) = \{y \in V : |y - x| \leq r\}$, then $|B_r(x)| \propto r^d$. For $\alpha = d$ (4.1) can then be interpreted as

$$\mathbf{P}(x \rightsquigarrow y) \propto \frac{1}{|B_r(x)|}. \tag{4.2}$$

This general principle, that under navigable augmentation the probability that $x$ links to $y$ should be inversely proportional to the number of vertices that are closer to $x$ than $y$ has been observed to hold across a wider class of graphs than just the grids, see e.g. [40] [22] [44], and seems to be the general technical requirement for navigability. It leads directly to our first construction.

A natural generalization to grids is to study graphs which are naturally grid-like. Let $B_r(v)$ be as above, but using graph instead of grid distance.

**Definition 4.2.1.** *A family of graphs has* bounded doubling size *if there exists a family wide constant $c$ such that for all $G = (V, E)$ in the family, $u, v \in V$, and $r \geq 1$*

$$B_r(u) \subset B_{2r}(v) \Rightarrow |B_{2r}(v)| \leq c|B_r(u)|.$$

This means that ball's sizes are polynomial in their radius, and in many cases the commonly used *doubling dimension* of a family roughly corresponds to the $\log_2$ of the smallest such $c$. Clearly, all common lattices have this property. It is not the widest class of graphs where navigable augmentation is possible, in fact [22] and [29] have shown that families with a sufficiently slowly growing dimension can still be made navigable, but it provides a good compromise between generality and convenience for us.

In the constructions below, we will augment the base graph with more than one long-range edge per vertex. In general, a $k$ edge augmentation is expected to give $O(\log^2 n/k)$ expected greedy routing time. Our constructions will generate close to $\log n$ edges per vertex in expectation[1], and thus have routing time $O(\log n)$. They remind most of previously explored finite long-range percolation models, for which the diameter is known to be $O(\log n/\log\log n)$ [17].

## 4.3 The Independent Interest Model

We start by introducing a conceptually simple model. Compared to our main model below, it is not a particularly interesting model of network dynamics, and not particularly realistic, but serves to illustrate the reasoning we will use later.

Let $X_1, X_2, \ldots, X_n$ be $n$ random variables drawn from an exchangeable joint distribution such that $\mathbf{P}(X_i = X_j) = 0$ for $i \neq j$. It is well known in this case that the probability that for any $k$, $\mathbf{P}(X_k \geq X_j \text{ for all } j < k) = 1/k$, and that this event is independent for each $k$. This fact, combined with (4.2) motivates the following model.

**Definition 4.3.1.** (The Independent Interest Graph) *Let $G = (V, E)$ be a graph, and for $x, y \in V$ let $d(x, y)$ be the graph (geodesic) distance between them.*

[1] A degree going to infinity may seem unrealistic in terms of social networks, but note that log(6 billion) $\approx 22.5$ which is probably considerably less than the average number of acquaintances a person has in the real world for most definitions of the word "acquaintance". Our models can be given bounded degree by simply thinning the edges (removing each edge independently with probability $1 - 1/\log(n)$).

*Create the long range links as follows: For each vertex, independently create an exchangeable sequence of random variables $(X_y^x)_{y \in V}$. The add an edge from $x$ to $y$ if:*

$$X_y^x \geq X_z^x \text{ for all } z \neq x \text{ s.t. } d(x,z) < d(x,y).$$

In the social network metaphor, each $X_y^x$ in the construction above can be seen as $x$'s interest in $y$, and the construction simply means that $x$ befriends each $y$ who is more interesting to him than any closer person. In other words, starting from his own position, $x$ searches outwards for friends, befriending each new person he meets if that person is more interesting to him then the people he already knows. In reality, of course, it is unlikely that the interest levels $X_y^x$ would be independent for each $x$ and $y$ – in particular, one would expect a high correlation between $X_y^x$ and $X_x^y$. This fact will inspire our later models below.

That the independent interest graph is navigable in fact follows from the observations above and previous results, but for illustration we will give a direct proof here.

**Theorem 4.3.2.** *For any family of connected graphs with bounded doubling size, the expected greedy path between any two vertices has expected length $O(\log n)$, where $n$ is the size of the graph.*

*Proof.* Let $z \in V$ be the target vertex. We follow the standard method: divide $V$ into $O(\log n)$ phases, with the $i$-th phase defined as the set of vertices $x$ such that $2^{i-1} < |x - z| \leq 2^i$. At a vertex $x$ in the $i$-th phase, for $i \geq \log \log n$, let $A$ be the event that $x$ has a shortcut to a lower phase, that is

$$A = \{x \rightsquigarrow y : y \in B_{2^{i-1}}(z)\} = \{x \rightsquigarrow B_{2^{i-1}}(z)\}.$$

Let

$$w = \underset{v \in B_{(3/2)2^i}(z)}{\operatorname{argmax}} (X_v^x)$$

By construction, $x$ has a link to $w$, so $A$ will occur if $w \in B_{2^{i-1}}(z) \subset B_{(3/2)2^i}(x)$. That the family has bounded doubling size thus means there is a constant $c$ such that $B_{(3/2)2^i}(x)/B_{2^{i-1}}(z) \leq c^2$. Since each vertex in the larger ball is equally likely to be the most interesting.

$$\mathbf{P}(A) \geq \frac{1}{c^2}$$

independent of $n$ and $i$. If $A$ does not occur, then in the next step we are by necessity not further from $z$ (nearest neighbors in base graph are always connected), and because the edges are chosen independently, $A$ occurs at the new vertex with at least the same probability. Therefore the expected number of steps until $A$ occurs, an upper bound on the number steps in a phase, is at most $c^2$.

For each sufficiently big phase, we thus have a constant bound on the expected number of steps. Since the destinations of the edges at each vertex are independent of the previous path taken by the query, it follows that the expected number of steps in such phases is at most the sum over all of them, which is $O(\log n)$. Only $O(\log n)$ points in smaller phases remain, so the result holds. □

## 4.4 The Double Clustering Model

Our main model of interest is conceptually similar to the independent interest model of the last section, but rather than letting each vertex' interest in each other vertex be an independent random variable, we let each vertex also live in a second space, and let the interest between two vertices be their proximity in that space. In a social network, this can be represented by each individual not only living somewhere in the physical world, but also having some position in a less clearly defined "space of interests" (his job, activities, etc.). People who live close to one another tend to become acquainted by "default", while people befriend those far away only if they have interests that agree to some extent.

In the constructed graph, each vertex is thus connected to every other vertex that is at least as "interesting" as any other which is at most as "far away". Formally, let a distance function be a real valued kernel $d(x, y)$ such that $d(x, x) = 0$ and $d(x, y) + d(y, z) \geq d(x, z)$ but which is not necessarily symmetric. The most general definition of such a graph is then:

**Definition 4.4.1.** (The Double Clustering Graph) *Let $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ be set two sets of points in possibly different spaces $M_1$ and $M_2$*

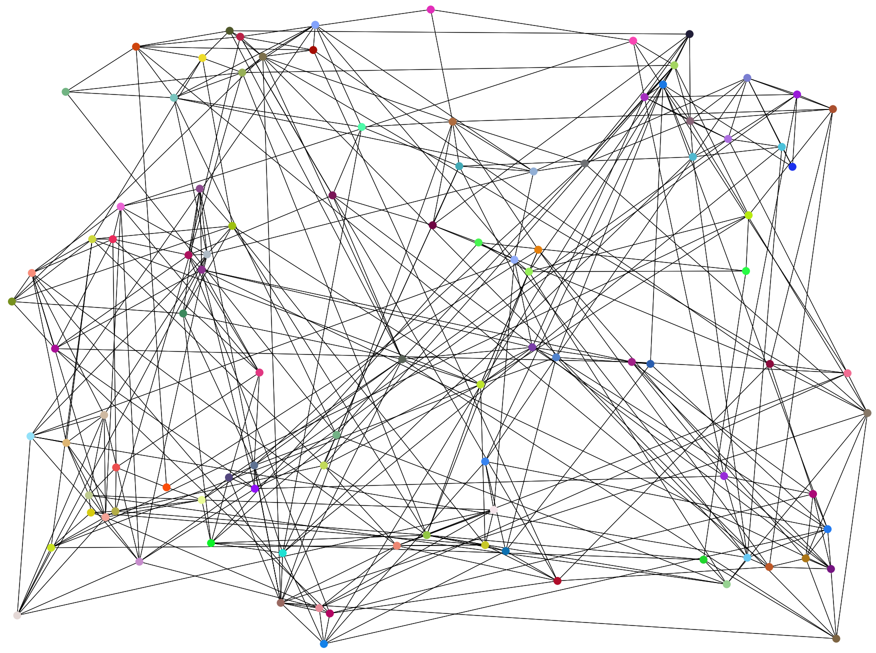


Figure 4.1: A double clustering graph of 100 vertices. Each vertex has a random position in a two-dimensional physical space ($[0, 1.33] \times [0, 1]$), as well as a in a three-dimensional color space (RGB) ($[0, 1]^3$), both using Euclidean distance.

*with distance functions* $d_1$ *and* $d_2$ *respectively. The graph* $G = (V, E)$ *is constructed as follows:*

- $V = \{1, 2, \ldots, n\}$.
- $(i, j) \in E$ *if for all* $k \in V$, $k \neq i, j$*:*

$$d_1(x_i, x_k) < d_1(x_i, x_j) \Rightarrow d_2(y_i, y_k) \geq d_2(y_i, y_j)$$

Note that the two sequences are symmetric in the definition, and that $G$ contains a nearest neighbor graph for both point sets. If, in particular, we let the $x_i$ and $y_i$ be the vertices of two graphs $G_1$ and $G_2$, letting $d_1$ and $d_2$ be graph distance, we may see the construction as an augmentation of either one to create a denser graph.

Since we are interested in probabilistic models, we want to let $(x_i)$ and $(y_j)$ be random points. One way of doing this is to let $\pi$ be a random permutation of $[n]$, and then letting $y_i = x_{\pi(i)}$. In the graph case, this corresponds to:

**Definition 4.4.2.** (Random Double Clustering Graph) *For a vertex set $V$, let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two given graphs. Let $\pi$ be a random permutation of $V$, and construct $G' = (V, E')$ by letting $(u, v) \in E'$ if for all $w \in V$, $w \neq u, v$:*

$$d_1(u, w) < d_1(u, v) \Rightarrow d_2(\pi(u), \pi(w)) \geq d_2(\pi(u), \pi(v))$$

*where $d_1$ and $d_2$ graph distances in $G_1$ and $G_2$ respectively.*

Note that every edge added in the construction has a direction, though in many cases (such as nearest neighbors in $G_1$ and $G_2$) edges in both directions will be included. One may choose to see the resulting graph as undirected by simply removing directionality and duplicated edges. For the sake of bounding the routing time, it is advantageous to preserve directionality and route using only outgoing edges.

In light of this, and before proceeding to analysis, we note that the construction $G'$ works equally well if $G_1$ and $G_2$ are directed graphs.

## 4.5 Analysis of Double Clustering

We will analyze special cases of Definition 4.4.2. We start by proving that greedy routing takes only $O(\log n)$ steps in expectation when we construct a double clustering graph using two directed cycles. Augmenting a directed cycle is the most basic form of Kleinberg type navigability, and has been extensively investigated in the case of independent augmentation (see e.g. [6]), but of course is not a good model for most real world scenarios.

More general models, in particular where the spaces are not directed, are more complicated. In these cases the probability of finding a link that halves the distance to the destination is not independent of the previous route taken. This can be seen in the simulations below, where double clustering graphs have slightly longer greedy paths than the equivalent independent interest graphs, though seemingly only by a constant. We attempt an analysis of one class of such models, where the first graph may take a more general form, and the second is an *undirected* cycle (in particular, this includes the case of two undirected cycles), but to do so we are forced to modify the routing used somewhat. The resulting algorithm is still a form of decentralized routing by Kleinberg's definition. Using this, we are able to show that routing takes a polylogarithmic number of steps, a somewhat worse bound than what we expect is true.

We conjecture that double clustering can be applied to just about any graph (see the conclusion), but can not yet prove it.

### 4.5.1 Two Directed Cycles

Let $G_1$ and $G_2$ in Definition 4.4.2 be two cycles of $n$ points, that is the directed graphs with vertex set $V_1 = V_2 = \{0, 1, \ldots, n-1\}$ and both $E_i$ containing an edge from $u$ to $u + 1$ (modulo $n$) for each $u \in V_i$. We will refer to this special case as the Double Cycle Graph. It constitutes the simplest case of double clustering.

Below, $d(x, y) = y - x \mod n$ will be graph distance in the cycles, and $d_\pi$ will be the corresponding distance function on the permuted cycle ($d_\pi(x, y) = d(\pi(x), \pi(y))$). We will discuss greedy routing using $d$, but by symmetry the same results hold for $d_\pi$.

Note from the definition that $G$ contains a link to the point $y$ such that $d(x, y) = 1$ (the next vertex in the cycle) and the point $z$ such that $d_\pi(x, z) = 1$ (the next vertex in the permuted cycle).

Addition of vertex values below is always modulo $n$, but the notation is suppressed for readability.

**Lemma 4.5.1.** *For $w \neq z \in V$, let $w' \in V$ be the vertex that $w$ routes to when $d$-greedy routing for $z$. Then:*

- *$w'$ lies inclusively between $w + 1$ and $z$ in the cycle (that is $d(w', z) < d(w, z)$).*
- *$w'$ lies inclusively between $w + 1$ and $z$ in the permuted cycle ($d_\pi(w', z) < d_\pi(w, z)$).*

*Proof.* The first statement is obvious from the definition of greedy routing, and the fact that $w \rightsquigarrow w + 1$ so there is always a choice which approaches $z$.

To prove the second statement, assume that $w'$ is not between $w+1$ and $z$ in the permuted cycle. This means that $d_\pi(w,z) < d_\pi(w,w')$. Let $A$ be the set of points inclusively between $w'+1$ and $z$ (that is $A = \{w'+1, w'+2, \ldots, z\}$). Define $q$ as the first point in $A$ such that $d_\pi(w,q) < d_\pi(w,w')$, noting that at least one such point, $z$, exists. By construction, and since $w \rightsquigarrow w'$, $q$ must be closer to $w$ in the permuted cycle than any vertex between $w$ and itself, and thus $w \rightsquigarrow q$. But if this were the case, $w$ would have routed to $q$ and not $w'$, which is a contradiction. □

**Corollary 4.5.2.** *For any permutation $\pi$, a $d$-greedy path from any vertex $y$ to any other $z$ in the double cycle monotonically approaches $z$ in $d_\pi$, likewise a $d_\pi$-greedy path monotonically approaches $z$ in $d$.*

In light of the corollary, it might seem that greedy routing with respect to $d$ and $d_\pi$ would produce the same paths. In fact, this is not the case, which we prove as an aside:

**Lemma 4.5.3.** *There exists a permutation $\pi$ such that greedy routing from some vertex $y$ to some vertex $z$ with respect to $d$ and $d_\pi$ produces different paths.*

*Proof.* Let $\pi$, $y$ and $z$ be such that there are exactly two vertices $x_1$ and $x_2$ that lie between $y$ and $z$ in the cycle ($d(y,z) > d(x_1,z) > d(x_2,z)$), and also lie between $y$ and $z$ in the permuted cycle. Let $x_1$, $x_2$ appear in the opposite order the permuted cycle ($d(y,z) > d_\pi(x_2,z) > d_\pi(x_1,z)$).

Note that by construction, $y$ will have edges to both $x_1$ and $x_2$ in the double cycle graph, because $x_1$ is closer in $d_\pi$ then any $d$ closer point to $y$, and likewise for $x_2$ (in particular, it is closer to $y$ than $x_1$ in $d_\pi$). However, when greedy routing with respect to $d$ for $z$, $y$ will choose $x_2$, while when greedy routing with respect to $d_\pi$, it will choose $x_1$. □

Marginally, under a uniform random choice of $\pi$, the probability that $x \rightsquigarrow y$ in the double cycle model is exactly $1/d(x,y)$ as it should be for navigability. However, like in the all the double clustering graphs, the random edges are not formed independently, so the situation is different from previous results. We will see, however, that in the double cycle, the monotonicity of the routing path also in $d_\pi$, as proved above, makes the routing events independent (in a sense which will be shown precisely below). The knowledge provided by previous routing steps is always "behind us" in the permuted cycle.

**Theorem 4.5.4.** *For any two points $y, z \in [n]$, the greedy path from $y$ to $z$ in the double cycle graph formed by a uniformly random permutation $\pi$ has expected length $O(\log n)$.*

*Proof.* The proof method is the same as in Theorem 4.3.2, thus we will consider starting in a point $x$ such that $r > d(x,z) \geq r/2$ and bound the expect number of steps (conditioned on the earlier path) until the route is within $r/2$ of $z$.

Divide the vertices between $x$ and $z$ in the cycle into two equal sized sets $R$ and $H$, so that if $d(x,z)$ is odd

$$R = \{x+1, x+2, \ldots, x + \frac{d(x,z)+1)}{2}\}$$
$$H = \{x + \frac{d(x,z)+1}{2} + 1, \ldots, z-1, z\}.$$

If $d(x,z)$ is even, we let $R$ end at $x + (d(x,z)/2)$ and $H$ go from there to $z-1$ so that $R$ and $H$ retain the same size.

Note that if $x \rightsquigarrow H$, then we can route to a point with distance to $z$ less than $r/2$, and that

$$\mathbf{P}(x \rightsquigarrow H) = \mathbf{P}(d_\pi(x,H) < d_\pi(x,R)) = 1/2$$

where $d_\pi(x,S)$ means the minimal distance from $x$ to any point in the set $S$.

Let $A$ be the event that $d_\pi(x,H) < d_\pi(x,R)$, and $B$ be the event that before reaching $x$ we greedy routed along the path

$$x_1 \rightsquigarrow x_2 \rightsquigarrow \ldots \rightsquigarrow x_k \rightsquigarrow x$$

for some $k$ and sequence of vertices where $d(x_i,z) < d(x,z)$. We will show that $\mathbf{P}(B \cap A) = \mathbf{P}(B \cap A^c)$, which (since $P(A) = P(A^c)$) implies that $\mathbf{P}(B \,|\, A) = \mathbf{P}(B \,|\, A^c)$ and thus that $A$ and $B$ are independent.

To do this, we define a bijection between the set of permutations $B \cap A$ and $B \cap A^c$. For a given $\pi \in B \cap A$, let $\pi'$ be $\pi$ composed with a permutation that flips the positions of the elements in $R$ and $H$. Clearly, if $\pi \in A$, then $\pi' \in A^c$.

By Corollary 4.5.2 $d_\pi(x_i,z) > d_\pi(x,z)$ for all the $x_i$ in the definition of $B$. This means that all the vertices in $R \cup H$ are further from each $x_i$ than $x$ in both $d$ and $d_\pi$. Thus the internal order of vertices in $R \cup H$

can not affect the edges of the $x_i$, and if $\pi \in B$, then $\pi' \in B$ as well. It follows that $|B \cap A| = |B \cap A^c|$, whence

$$\mathbf{P}(A \,|\, B) = \mathbf{P}(A) = 1/2$$

for any $B$ defined as above. At each vertex we reach at distance between $r$ and $r/2$ to $z$, the probability of having a link to a vertex with distance less than $r/2$ is thus greater than $1/2$ regardless of which vertices we visited previously. The result now follows as in Theorem 4.3.2. □

### 4.5.2 Bounded Doubling Size and an Undirected Cycle

In this section, we let $G_1$ belong to a more general family meeting the criteria of Definition 4.2.1, and we let $G_2$ be an undirected cycle (a one-dimensional toric grid). Like in previous cases, we shall bound the expected number of steps that it takes to halve the distance to the destination: however, unlike in previous cases, the event of halving the distance in each step of greedy routing is not independent of the previous path.

In order to control the dependencies between the edges encountered at each step, we introduce a modified routing algorithm we call *half-greedy routing.* When routing for a vertex $z$ and currently at $x$, we examine each of $x$'s neighbors in the double clustering graph $G$. If any neighbor $w$ is such that $d_1(x,z) > 2d_1(w,z)$, then $w$ is chosen for the next step. If no such such $w$ is found, $x$ routes to a neighbor $w'$ in $G_1$ such that $d_1(w',z) = d_1(x,z) - 1$ (choosing from all possible such $w'$ by some deterministic rule).

Half-greedy routing thus either takes a "very big step", which immediately halves the distance, or a very small step to the next vertex in $G_1$. Intuitively, one may imagine this as a participant in a Milgram style experiment only bothering to send the letter by post if he knows somebody very suitable, and otherwise just giving it directly to one of his neighbors. The analytical advantage of this approach is that while subsequent vertices reached by a greedy route do not have independent positions in $G_2$, neighbors in $G_1$ (nearly) do. The navigability result thus follows from this lemma:

**Lemma 4.5.5.** *Let $\pi$ be a random permutation of $[n]$ and $d_\pi$ be circular* *distance under this permutation. That is, for $x, y \in [n]$*

$$d_\pi(x, y) = \min(|\pi(x) - \pi(y)|, n - |\pi(x) - \pi(y)|)$$

*Let $A$ and $B$ be disjoint subsets $[n]$, such that $|A| = k$ and $|B| \geq qk$ for some $q > 0$. The elements of $A$ are enumerated $a_1, a_2, a_3, \dots, a_k$. Define a random variable $\tau$ by*

$$\tau = \min(t \geq 0 : d_\pi(a_t, A\backslash\{a_t\}) \geq d_\pi(a_t, B))$$

*or $\tau = k$ if this never occurs. Then, for $t < k/5$*

$$\mathbf{P}(\tau \geq t) \leq e^{-mt}$$

*where $m = m(q) < \infty$, a constant independent of $n$ and $k$.*

We will establish this lemma below. First we show how it leads to the desired result.

**Theorem 4.5.6.** *In Definition 4.4.2 let $G_1$ be a connected graph from a family with bounded doubling size, and $G_2$ be an undirected cycle. Then the path though $G$ between any two vertices $x$ and $z$ when half-greedy routing with respect to $d_1$ has expected length $O(\log^2 n)$.*

*Proof.* Let $T$ be the time it takes for half-greedy routing between any two vertices. We will establish the stronger fact that for $n$ sufficiently large and a constant $h$

$$\mathbf{P}(T \geq h(\log n)^2) \leq \frac{\log_2 n}{n}. \tag{4.3}$$

It then follows that

$$\begin{aligned} \mathbf{E}[T] &\leq h(\log n)^2 \left(1 - \frac{\log_2 n}{n}\right) + n\frac{\log_2 n}{n} \\ &= O(\log^2 n). \end{aligned}$$

Fix a destination $z$, and let the phases be as in the proof of Theorem 4.3.2. Consider the $i - th$ phase (the set of vertices $x$ such that $2^{i-1} < d_1(x, z) \leq 2^i$), where $i$ is such that the phase is "big", meaning it contains more than $\log^2 n$ vertices. We let $A$ and $B$ from Lemma 4.5.5 be defined by

$$B = B_{2^{i-2}}(z)$$

and

$$A = B_{5(2^{i-1})}(z) \backslash B$$

where the $B_r(z)$ are balls with respect to $d_1$. We note that (distance below always means $d_1$ except where otherwise noted):

1. Each vertex in the $i$-th phase belongs to $A$.

2. All vertices in $B$ are within distance $3(2^{i-1})$ of any vertex in the $i$-th phase.

3. Every vertex within distance $3(2^{i-1})$ of a vertex in the $i$-th phase is in $A \cup B$.

Together, these three facts mean that if a vertex $x$ in $i$-th phase has a randomly assigned position in $G_2$ (the cycle) which is at least as close (with respect to the permuted positions in $G_2$) to a vertex in $B$ as any vertex other than itself in $A$, the resulting double clustering graph $G$ will have an edge from $x$ into $B$.

Now consider half-greedy routing starting from a vertex $x$ in the $i$-th phase. Let the enumeration of $A$ be so that $a_1 = x$ and each subsequent $a_j$, for $j$ up to some $\ell$, is the vertex where $a_{j-1}$ would route if a "very big step" was not found. $a_\ell$ is the first vertex encountered so that it has a $G_1$ neighbor in a lower phase, after this we may order the elements of $A$ as we wish.

Since each vertex in $B$ is less than half as far from $z$ as those in phase $i$, the random variable $\tau$ from Lemma 4.5.5 thus dominates the time we spend in the $i$-th phase after starting from a given vertex.

Let $b = 2/m$, where $m$ is the constant from Lemma 5.5 with $q = 1/c^4 \geq |B|/|A|$ because of the bounded doubling size. Note that $q$, and thus $m$ and $b$, are independent of which phase we are in. Let $E^i_x$ be the event that we spend more than $b \log n$ steps in the $i$-th phase after starting from a vertex $x$ in the phase. Lemma 4.5.5 and the argument above gives

$$\mathbf{P}(E^i_x) \leq \frac{1}{n^2}.$$

Since the probability is simply uniform measure of permutations of $[n]$, this means that starting for any given vertex $x$ in the phase, routing to the next phase will take more than $b \log n$ steps in less than $1/n^2$ of all the permutations. Since the graph is dependent, where we enter the phase may depend on the permutation, but the very worst case scenario is that we always enter the phase at the vertex where it will take the most steps to route to the next. Let $E^i$ be the event that starting from *any* vertex in the $i$-th phase, we spend more than $b \log n$ steps in the phase.

$$\begin{aligned} \mathbf{P}(E_i) &= \mathbf{P}\left(\cup_{i\text{-th phase}} E_x^i\right) \\ &\leq \sum_{i\text{-th phase}} \mathbf{P}(E_x^i) \leq \frac{1}{n} \end{aligned}$$

where the last inequality holds because every phase trivially contains at most $n$ vertices.

There are at most $\log_2 n$ “big” phases, so the probability of spending more than $b \log n$ in any of them is less than $\log_2 n/n$ by another union bound. Since the number of vertices in the “small” phases is less than $4 \log^2 n$, (4.3) follows with $h = b/\log(2)$. □

The remainder of this section is a proof of Lemma 4.5.5. In order to establish the Lemma, we will make use of something we call the *toy train track* construction of a random permutation. We equate each vertex on the cycle with a curved segments of track in toy train set. These segments can be attached to each other to make longer sections[2], and when all the $n$ segments are attached they form a complete circle. All the pieces start in a bin, and are either red (corresponding to vertices in $A$), blue (corresponding to vertices in $B$), or gray (corresponding to vertices in neither set). We build the random circular track, starting as follows:

1. We pick up the segment of track corresponding to $a_1$ from bin, this is our current section.
2. Uniformly select from the remaining pieces a segment $x$ to attach clockwise from the current section, and then another segment $y$ to attach counterclockwise from the section.

[2] Our chosen vocabulary is to consistently use *segment* for each element, and *section* for connected collections of segments.

3. As long as neither the $x$ nor $y$ picked up in the last step is a blue or red piece, we continue to draw two new pieces to attach to the section.

This continues until a red or blue segment has been attached at one or both ends of the section. At this time the first construction stage is completed, and we *put the constructed section of track back in the bin* together with the other pieces. If at least one end was blue, then the building phase terminates.

If no blue piece was found, we start the second construction stage, we try to take out $a_2$ from the bin. If $a_2$ cannot be found on its own (it was part of the previous section), then the stage ends immediately. If it is found, then we proceed to build a new section starting from it as in the first stage, but this time we stop whenever a blue segment, a red segment, or the previously constructed section of track is attached to $a_2$'s section. At the end of stage two, we put $a_2$'s section back in the bin as before (if one was built), and, unless a blue piece was found, continue to stage three, which we complete in a similar manner.

If at any time all the pieces have been added to one section the building phase terminates, and likewise if we run out of red pieces to start from. When the building phase has terminated, we attach all the sections and segments in the bin in a random permutation (draw one at a time, and attach clockwise from the last) to form a completed circle.

Let $X$ be the number of construction stages. We make three claims about this construction which together establish Lemma 4.5.5:

1. The circle of track segments created is a uniformly random circular permutation.

2. $X \geq \tau$ (as defined above) for the corresponding permutation.

3. For $t < k/5$, $\mathbf{P}(X \geq t) \leq e^{-mx}$, where $m$ depends on $q$ but not $k$ and $n$.

*Proof of Claim 1:* This follows from the conditional distribution of random permutations. If one conditions on two segments $s_1$ and $s_2$ being next to each other, then resulting is distribution is a random permutation of the remaining segments, with the $s_1s_2$ section uniformly inserted. This is equivalent to the returning of the section to the bin.

Likewise, another section $s_3s_4$ would simply be uniformly inserted again. The claim follows from a series of such arguments.

□

*Proof of Claim 2:* This is almost immediate. If we encounter a blue piece during construction stage $i$, then all the segments closer to $a_i$ than that piece were gray, hence $d_\pi(a_i, A\backslash\{a_i\}) \geq d_\pi(a_i, B)$. If we don't find a blue piece in any construction stage, then $X = k$ which is an upper bound on $\tau$.

□

*Proof of Claim 3:* Let $E_i$ be the event that we encounter a blue piece in the $i$-th construction stage. $X$ is $\min(i : E_i \text{ occurs })$ or $k$ if this is undefined. In the first stage, there are $k + qk$ pieces for which we terminate, and $qk$ are blue, so $\mathbf{P}(E_1) \geq q/(1 + q) =: p$ (in fact greater).

Conditioned on $E_1$ not occurring, we let $e_1$ and $e_2$ be the two end pieces, and note that

$$\mathbf{P}(E_2 \,|\, E_1^c) = \mathbf{P}(E_2 \,|\, E_1^c \text{ and } a_2 \neq e_1, e_2)\mathbf{P}(a_2 \neq e_1, e_2 \,|\, E_1^c).$$

If $a_2 \neq e_1$ or $e_2$, then second construction stage could proceed. However, since $E_1$ did not occur, this means that we removed at least two red segments from the bin, and added only one new terminating section. Thus:

$$\mathbf{P}(E_2 \,|\, E_1^c \text{ and } a_2 \neq e_1, e_2) \geq \mathbf{P}(E_1) \geq p.$$

We now have to lower bound $\mathbf{P}(a_2 \neq e_1, e_2 \,|\, E_1^c)$. The worst case is that both $e_1$ and $e_2$ are red, in which case we drew 2 red segments out of $k - 1$ possible.

$$\mathbf{P}(a_2 \neq e_1, e_2 \,|\, E_1^c) \geq 1 - \frac{2}{k-1}$$

Using similar arguments (and rather conservative estimates), it follows that for $i \leq k/5$

$$\mathbf{P}(E_i \,|\, E_1^c, E_2^c, \ldots, E_{i-1}^c) \geq p\left(1 - \frac{2(i-1)}{k-(i-1)}\right) = p\frac{k-3(i-1)}{k-(i-1)} \quad (4.4)$$

whence

$$\begin{aligned}\mathbf{P}(X \geq t) &= \prod_{i=1}^{t} \mathbf{P}(E_i^c \,|\, E_1^c, E_2^c, \ldots, E_{i-1}^c) \\ &\leq \prod_{i=1}^{t} \left(1 - p\frac{k-3(i-1)}{k-(i-1)}\right) \\ &\leq \left(1 - \frac{p}{2}\right)^t = e^{-mt}\end{aligned}$$

where $m = -\log\left(1 - \frac{p}{2}\right)$.

□

## 4.6 Simulations

Simulations support the conjecture that double clustering creates navigable graphs over a larger span of structures. In cases where the first graph is not a directed cycle, one can see that double clustering gives slightly worse performance than when the edges are independent, as is expected. However, the simulation data still strongly indicates a logarithmic growth of path-length with the size of the graph.

### 4.6.1 Combined Greedy Routing

Since the double clustering construction is symmetric, it should create equally navigable networks with regard to both spaces. Thus we can perform greedy routing in the double clustering graph with respect to either distance function (which will sometimes lead to different results, see below.)

A direct consequence of this is that we may try to route with respect to to both distance functions, using at each step that which seems most profitable. As above, assume that $z$ is the target of the route.

- At vertex $x$, we calculate $m_1 = d_1(w_1, z)$, where $w_1$ is the neighbor of $x$ which minimizes this. Similarly, calculate $m_2 = d_2(w_2, z)$.

- Let $n_1$ be the number of vertices within $m_1$ of $z$ in the first space ($M_1$) – if the space if homogeneous this is the volume of a ball

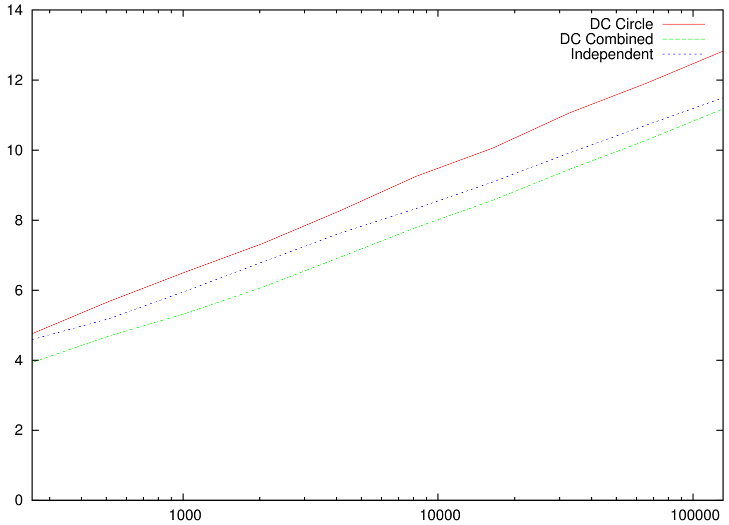


Figure 4.2: Performance of the double clustering graph versus the independent interest model using two undirected cycles.

of diameter $m_1$. Let $n_2$ be the equivalent for $m_2$ and the second space.

- Route to $w_2$ if $m_2$ is smaller than $m_1$, otherwise $w_1$.

We simulate combined routing as well as normal greedy routing for the models below. In these models it seems that the benefit of using this method regains that lost by the dependencies in the double clustering construction: combined greedy paths are shorter than greedy paths in the independent interest model of the same size.

### 4.6.2 Two Undirected Cycles

The simplest undirected double clustering model is the case of Definition 4.4.2 where both $G_1$ and $G_2$ are undirected cycles. A bound on half-greedy routing in this model is derived above, but we can simulate also the normal greedy algorithm. The results illustrated in Figure 4.2 – at all sizes simulated greedy routing with respect to either cycle produces slightly longer paths than the equivalent independent model,

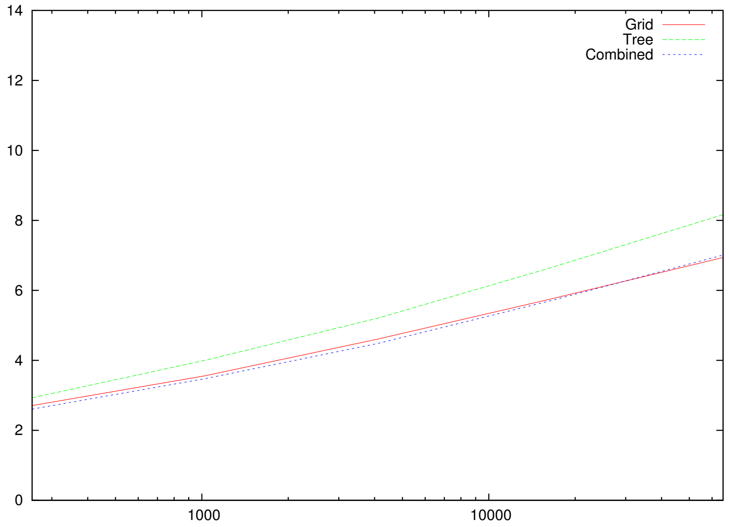


Figure 4.3: Performance of double clustering letting the first space (the geography) be a two dimensional grid, and the second a tree with the points as leaves (a categorization).

while combined greedy routing produces slightly shorter paths. All lines seem to follow strictly logarithmic growth.

### 4.6.3 A Grid and a Tree

Kleinberg's original work started with a two dimensional grid as the base graph and distance function, inspired, one expects, by the dimensionality, if not population distribution, of the surface of the earth. Later he [40] and Watts et al. [61] proposed equivalent models based on letting vertices have positions at the leaves of a tree. The tree represents a hierarchical model of information, ideas, interests or other characteristics, and the distance function is standard tree distance: $d(x, y)$ is the depth of smallest subtree containing both $x$ and $y$. The criteria for navigable augmentation in these cases is consistent with (4.2).

A natural attempt at a realistic double clustering model is to combine both of Kleinberg's models – we let the first space be a grid, and the second be a hierarchical tree structure (in our case, a binary tree, though any other branching is possible). We note that while the tree distance provides a well defined metric, this space can not be seen as a graph, so this is a sub-model of Definition 4.4.1 rather than Definition 4.4.2. A problem with the more general model is that greedy routing is not necessarily always successful: we may reach a vertex other than the destination with no neighbor which is closer to the destination than itself. This can occur in this model when routing with respect to the tree, or the combined distance, but not in the grid (where links to neighbors in all directions always exist) – in our simulations we simply fail and discard such routes[3].

Figure 4.3 shows a simulation of this situation. Routing purely using the tree shows slightly worse performance than routing using the grid, and as such the advantage of the combined model is less than above (for the largest data-point simulated it was, in fact, nonexistent). As expected not all routes were successful – at a network size of $2^{16}$ about 0.8 of the routes using only the tree, and 0.9 of the routes using the combined routing were successful. Another effect of the tree is that the degree of the double clustering graph is much higher (since many vertices have the same distance, and we only require them to be as close as any previous.)

### 4.6.4 Continuum Models

Discrete and grid based models cannot realistically describe most naturally occurring networks: especially social networks which are characterized by individuals placed randomly in continuums and often with heterogeneous population density. Continuum models for navigable networks have been explored by Franceschetti and Meester [30] and [21] as well this author [55], and Liben-Nowell et al. [44] has proposed a model based on real data that includes a non-uniform Poisson density of positions. Figure 4.1 shows a simulation a continuum double clustering model with 100 vertices.

[3] In this case, when routing for $z$, we do allow $x$ to route to a vertex at the same distance from $z$ as itself if it has no better choice. So as to not cause loops, we forbid routing to a vertex already in the path. This is important since tree distance has the property that there are a very large number of vertices at the same distance from any other. A large majority of the routes fail when routing for tree distance if this is not allowed.

## 4.7 Conclusion

We have introduced a new graph construction, which when combined with a random permutation of the points used to create the graph gives rise to networks with navigable properties. These graphs are constructed from a single natural principle, and may help explain why graphs of this type occur in real world networks.

While we have established navigability under a several cases, the analysis presented here is far from complete. In a sense it is unfortunate that we are able to analyze an unrealistic model (the directed cycle) for an intuitive clear routing principle, while the proof for the more realistic model requires somewhat contrived routing. Theorem 4.5.6 also has an extra $\log n$ multiple included for technical reasons in the proof: we believe strongly that neither this term (the actual bound is $O(\log n)$) nor the use of half-greedy routing is actually necessary. In fact, based on the absence of any opposing evidence in simulations or otherwise, we believe

**Conjecture 4.7.1.** *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two families of graphs with bounded doubling size (not necessarily with the same constants). For any two graphs $G_1 \in \mathcal{F}_1$ and $G_2 \in \mathcal{F}_2$ of size $n$, the double clustering graph from Definition 4.4.2, allows greedy routing in $O(\log n)$ expected steps.*

Proving this in general is difficult since the structure of the two base graphs control the dependence between the edges in the construction. We are however hopeful that progress can be made in this direction. Making rigorous stronger conjectures about Definition 4.4.1 is also difficult since monotonic greedy paths between vertices may not always exist, but we believe that the resulting graph will be navigable whenever such augmentation is possible.

Beyond this, the double clustering graph, as a new form of graph construction, has not been analyzed for questions other than navigability. Questions such as connectivity, diameter, and edge length remain open in some or all cases. And, finally, the question of how well double clustering actually matches the real world has not been investigated.

## Acknowledgments

Thanks to my advisors Olle Häggström and Devdatt Dubhashi for helpful comments and corrections. Thanks also to Elchanan Mossel for comments that helped me complete the final proof.

# Chapter 5

# Phases Transitions and Large Clusters

## 5.1 Random Graphs and Finite Percolation

The $\mathcal{G}(n, p)$ family of random graphs, originally due to Erdős and Rényi, is constructed by letting the vertex set $V = [n] = \{0, 1, \ldots, n-1\}$, and letting every possible edge $\{u, v\}$ belong to the edge set $E$ independently with probability $p$. Bernoulli bond percolation models, on the other hand, are typically constructed by starting with a finite degree lattice and retaining only the edges therein in the same manner, and not any others (retained edges are called open).

The study of random graphs is thus the study of models without an underlying geometry, whereas percolation models depend heavily on the geometry and structure of the lattice on which they are defined. A reasonable question is to ask what happens if one relaxes the influence of the structure in percolation models – for example by allowing open edges to form between vertices more then one step from each other in the lattice. Such models, known as *long-range percolation*, have been studied previously (see [50] [4] and below for more references).

Likewise, starting from the other direction, one may ask what happens to $\mathcal{G}(n, p)$ like graphs when structure is introduced – making some possible edges more likely to appear than others. Results about this can be gleaned from recent generalized random graph models [12], and show how much structure can be introduced while keeping the behavior of the model more or less intact.

In fact, in terms of the distance dependence of the edges, known models for long-range percolation and generalized random graphs come very close to covering the whole spectrum. We will discuss a family of random graph models with a single real parameter $\alpha$ that regulates the influence of an underlying structure. We will see that the cases when $\alpha < 1$ fall in the category of previously analyzed random graphs, while the cases where $\alpha > 1$ fall in the category of long-range percolation models. We present some connectivity results for the final, critical, case where $\alpha = 1$.

### 5.1.1 Notation

As is common, we will use $G$ to denote both specific graph realizations and the random graphs, though we strive to make the difference clear by context. Where $\mathcal{G}$ is a random graph family, $G \sim \mathcal{G}$ means that $G$ is distributed according to this family. $C_1 = C_1(G)$ will denote the biggest connected component of $G$.

As usual, a series of events $A_1, A_2, \ldots$ occurring *asymptotically almost surely* (a.a.s.) means that $\lim_{n\to\infty} \mathbf{P}(A_n) = 1$.

### 5.1.2 Organization

The paper is organized as follows. In Section 5.2 we introduce our "$\alpha$-model" of random graphs, and in Section 5.3 we go through how the behavior of the model varies, discussing the known results for most values of $\alpha$. Finally Section 5.4 contains our analytic contribution.

## 5.2 The $\alpha$-model

A major difference between the analysis of random graphs and percolation models is whether it is done in a finite or infinite setting. Questions about percolation are typically asked about the behavior of clusters on a infinite grid – the most basic question being whether an infinite cluster remains open. $\mathcal{G}(n, p)$ is, on the other hand, almost always studied for finite values of $n$ - this for the simple reason that if $n = \infty$ and $p > 0$ the graph is a.s. not locally finite. The typical approach is instead to scale the value of $p$ with $n$ – in particular, sparse random graphs are ones where $p = c/n$ for some fixed $c$ (meaning that expected degree is essentially constant for all $n$). The question is then, rather than asking whether an infinite cluster exists, to look at the relative size of the largest cluster (compared to $n$) as a function of $c$.

We take the latter approach here, using a degree normalizer and studying finite graphs, but note that it largely intersects with normalizer free models in cases where the degree is already limited by the structure. We will also restrict ourselves to a one dimensional geometry. One dimension is not an interesting environment for standard Bernoulli percolation, but long-range percolation can be fruitful here. Our geometry is based around the following metric:

$$d(u,v) = \min(|u-v|, n-|u-v|).$$

This is equivalent to placing the vertices in a ring and using the geodesic distance (see Figure 5.1).

**Definition 5.2.1.** (The $\alpha$-model) *For $\alpha \in [0,\infty]$ the family of random graphs $\mathcal{G}_\alpha(n,c)$ are graphs $G = (V,E)$, where $V = [n]$ and for $u, v \in V$*

$$p_{u,v} = \mathbf{P}(\{u,v\} \in E) = \frac{c}{h_{\alpha,n} d(u,v)^\alpha}$$

*where $h_{\alpha,n} = \sum_{u \in V : u \neq 0} 1/d(u,0)^\alpha$ , independently for all disjoint $\{u,v\}$.*

For $\alpha = 0$ this equivalent to $G(n,p)$ with $p = c/(n-1)$. When $\alpha = \infty$

$$p_{uv} = \begin{cases} c/2 & \text{if } d(u,v) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

which is standard percolation, for which an infinite cluster cannot exist if $c < 2$.

Random graphs with edge probabilities given by a power-law of the distance are not new, and have appeared in more or less exactly this form elsewhere. See the pioneering work of Aizenman, Newman, and Schulman [50] [4], the ideas of Kleinberg [39], as well as later work by other authors [7], [17].

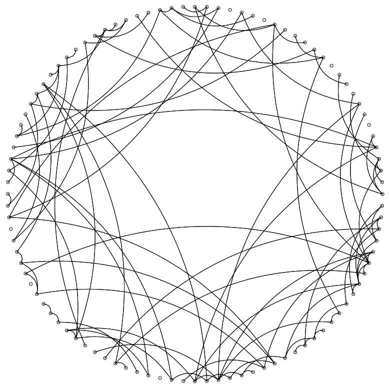


Figure 5.1: A realization of the $\alpha$-model $\mathcal{G}_\alpha(n,c)$ with $\alpha = 1$, $n = 100$, and $c = 2$.

## 5.3 Regimes of the $\alpha$-model : The Emergence of Structure

### 5.3.1 $\alpha = 0$ : $\mathcal{G}(n,p)$ Random Graph

When $\alpha = 0$, the distance between the points does not affect connectivity, and the $\alpha$-model is exactly the same as $\mathcal{G}(n,p)$ with $p = c/(n-1)$. This case has, of course, been extensively studied, see [26] as well as the book length discussions in [37] [10]. With regard to connectivity, it is known to undergo a phase transition at $c = 1$: the largest connected cluster is of size $\theta(\log n)$ in the subcritical phase, $\theta(n^{2/3})$ in the critical phase, and $\theta(n)$ in the supercritical phase.

**Theorem 5.3.1.** (Erdős, Rényi) *Let $G \sim \mathcal{G}(n,p)$ for $p = c/(n-1)$, and $\rho$ be the survival probability of a Galton-Watson branching process with Poisson$(c)$ offspring distribution. Then*

$$|C_1|/n \xrightarrow{p} \rho$$

*as $n \to \infty$.*

Much more is known regarding the distribution sequence of component sizes, see the above books for details.

### 5.3.2 $0 < \alpha < 1$ : Essentially a Random Graph

When $\alpha > 0$ the geographic structure of the model starts affecting the edges. However, as long as $\alpha < 1$ much of the general behavior is retained. While the random graph results above cannot be directly applied, this regime falls within a more recent general random graph model of Bollobás, Janson, and Riordan [12].

In the BJR model $\mathcal{G}(n, \kappa)$, one is given a "ground space" $\mathcal{S}$ which we take to be $[0, 1]$, and a Borel measurable kernel $\kappa : S \times S \mapsto \mathbb{R}^+$. For a sequence of points $(x_1, x_2, \ldots, x_n)$ of $\mathcal{S}$ corresponding the $n$ vertices, edges are added independently between each pair with probability:

$$p_{uv} = \min(\kappa(x_u, x_v)/n, 1).$$

The model contains a lot of freedoms which we will not require. For instance the sequence of points may be random, in which case the limiting distribution obviously matters greatly. We will not need this, and indeed may set $x_u = u/n$. We then let:

$$\kappa(x, y) = \begin{cases} c/|x-y|^\alpha & \text{for } x \neq y \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where $|x - y|$ is again interpreted as circular distance, this time on $[0, 1]$.

In order for the results of BJR to hold, $\kappa$ must adhere to certain conditions which the authors call being "graphical" (Definition 2.7 in [12]). The troublesome condition for our $\kappa$ is that it must belong to $L^1(S \times S, \mu \times \mu)$ (where $\mu$ in our case is the Lebesque measure on $[0, 1]$). Clearly this is true only for $\alpha < 1$.

For this case however, the BJR model with the above kernel gives $p_{ij} = c/(n^{1-\alpha} d(i, j)^\alpha)$ which is asymptotically equivalent to the $\alpha$ model but for a slightly different value of $c$. BJR prove that $\mathcal{G}(n, \kappa)$ behaves more or less like the classical $\mathcal{G}(n, p)$ – in particular, most of the proofs are very similar barring the technical difficulties incurred by the greater generality. With regard to connectivity, the same result as above holds, that the phase transition at which a $\theta(n)$ cluster emerges is at $c = 1$.

**Theorem 5.3.2.** (Bollobás, Janson, Riordan) *If $G$ is a random graph of size $n$ from the BJR model with kernel $\kappa$, then*

$$|C_1|/n \xrightarrow{p} \rho_\kappa$$

*where $\rho_\kappa$ is the survival probability of a multitype Galton-Watson process with the offspring distribution of $x$ given by a Poisson point process on $\mathcal{S}$ with intensity $\kappa(x, \cdot)$.*

Because our geometries are transitive, this reduces to a single-type Galton-Watson process with Poisson offspring. It is thus well known that $\rho > 0$ exactly when the expected number of offspring is greater than 1.

### 5.3.3 $\alpha = 1$ : The Small World

When $\alpha = 1$, the resulting model no longer falls within the that of BJR since the $\kappa$ implied by (5.1) is no longer integrable at 0. On the other hand, it does not fall within the long-range percolation models discussed below, this time since $1/x$ is not integrable at infinity. The connectivity of the resulting "in-between" model has to our knowledge not been studied elsewhere.

This model is of added interest due to the results of Kleinberg regarding small-world models and further work that has followed, see [39] [41]. These show that the value of $\alpha$ is intimately related with possibility of decentralized routing (path-finding) in graphs, and that it is exactly when the relation between distance and edge prevalence is as at $\alpha = 1$ that such routing will find short paths.

In Section 5.4 below, we prove that this case also undergoes at phase transition equivalent to the previous – when the expected degree of each vertex is greater than one, $C_1$ has order $n$, otherwise it is sublinear.

### 5.3.4 $1 < \alpha < 2$: Long-Range Percolation

When we move to $\alpha > 1$ the model undergoes another radical change. In this case the normalizing constant $h_{\alpha,n} \to h_\alpha < \infty$ as $n \to \infty$. The model is thus very similar to the long-range percolation models studied in the 1980s within mathematical physics (see [50] and [4] for rigorous results). It was found that these models do undergo a phase transition similar to the regimes above: for $c$ less than a certain value there is no percolation, while for large $c$ it can be shown to occur.

While the theorem is stated for percolation on $\mathbb{Z}$ in [50], the arguments in the proofs use only finite subsets, and may be stated as

**Theorem 5.3.3.** *(Newman, Schulman) For* $1 < \alpha < 2$ *there exists* $c < \infty$ *and* $\phi > 0$ *such that for* $\mathcal{G}_\alpha(n, c)$

$$|C_1| \geq \phi n \quad a.a.s.$$

In all the regimes of the $\alpha$-model discussed in Sections 5.3.1 – 5.3.3, the critical value of $c$ was found to be one. This is not possible when $\alpha > 1$.

**Proposition 5.3.4.** *For any* $\alpha > 1$*, there exists an* $\epsilon > 0$ *such that for* $c = 1 + \epsilon$ *in* $\mathcal{G}_\alpha(n, c)$ $|C_1|/n \to 0$ *a.a.s.*

*Proof.* Let $\epsilon = 1/(6h_\alpha^3)$. For a given graph, let $K(v)$ be the clustering number of a vertex $v$. That is

$$K(v) = \text{number of 3-cycles containing } v$$

For a given vertex $v$, let $Y$ be the number of second order neighbors.

$$Y \leq \sum_{u \in N(v)} (|N(v)| - 1) - K(v)$$

which implies

$$\mathbf{E}[Y] \leq \mathbf{E}[|N(v)|]\mathbf{E}[|N(u)|] - \mathbf{E}[K(v)].$$

Since $\mathbf{E}[K(v)] \geq \mathbf{P}(v \leftrightarrow v+1, v \leftrightarrow v+2, v+1 \leftrightarrow v) = 1/(2h_\alpha k^3)$, it holds that $\mathbf{E}[Y] < 1$. It follows that any two-step exploration process on the graph is dominated by a subcriticial Galton-Watson process. □

### 5.3.5 $\alpha = 2$ : The Second Critical Value

The situation when $\alpha = 2$ is also handled in [50] and [4]. Here it turns out that for the $\alpha$-model, as we have defined it, there is no giant cluster unless $c$ is large enough that $p_{u,u+1} = 1$ (in which case everything is of course trivially connected). However, the authors show that there are in fact distributions with this tail decay where you will get a giant cluster, however, it is necessary that:

$$\liminf_{d \to \infty} d^\alpha p_{u,u+d} \geq 1$$

In our case, this implies that $c = h_\alpha$, which of course is the trivial case where $p_{i,i+1} = 1$. For more general long-range percolation formulations one can define non-trivial situations which do percolate with this tail.

### 5.3.6 $2 < \alpha < \infty$ : Essentially Percolation

When $\alpha > 2$, the model shows behavior similar to standard percolation. A way to see this is to consider a renormalization of the vertex space into large blocks. For $c < h_\alpha$ let $B_1, B_2, \ldots, B_{n/m}$ be contiguous blocks of vertices, each of size $m$. Let $X_{i,j}$ be the number of edges between blocks $i$ and $j$.

$$\mathbf{E}[X_{i,i+1}] \leq \sum_{x=-\infty}^{0} \sum_{y=1}^{\infty} \frac{1}{h_\alpha |x-y|^\alpha} \leq \infty.$$

Since $X_{i,i+1}$ is the sum of independent events each occurring with probability smaller than one, that it has bounded expectation means that

$$\mathbf{P}(X_{i,i+1} > 0) \leq p < 1$$

where $p$ is independent of $n$ and $m$. On the other hand, if $|i - j| > 1$, then

$$\mathbf{P}(X_{i,j} > 0) \leq \mathbf{E}[X_{i,j}] < h'/m^{\alpha-2} \to 0$$

as $m \to \infty$ ($h'$ is a constant).

This means that if $m$ is sufficiently large, and we view two blocks as connected if there is any edge between them, then the system of connected blocks will look more or less like standard Bernoulli percolation.

Formally, let $k = \log n$ and $m = (\log n)^{3/(\alpha-2)}$. Then the integral bounds give that for any vertex $x$

$$\begin{aligned}&\mathbf{P}(x \text{ is connected to at least } mk \text{ vertices}) \leq \\ &\quad \mathbf{P}(k \text{ adjacent blocks are connected in either direction from } x) \\ &\quad + \mathbf{P}(\text{one of those } 2k \text{ blocks has a non-adjacent connection}) \leq 1/\log n.\end{aligned}$$

for $n$ sufficiently large. It follows that $\mathbf{E}[\#\text{ vertices in clusters larger than } mk] < n/\log n$ whence $\mathbf{P}(|C_1|/n > \rho) \to 0$ for any $\rho > 0$.

### 5.3.7 $\alpha = \infty$ : Percolation

As noted above, at $\alpha = \infty$ the $\alpha$-model is exactly percolation with probability $c/2$ that each edge is open.

## 5.4 Analysis of case $\alpha = 1$

In this section, we prove a result similar to Theorems 5.3.1 and 5.3.2 for the "Small World" case where $\alpha = 1$.

To $\mathcal{G}_1(n, c)$ we associate a Galton-Watson branching process $\{Z_t^n(c)\}_{t \in \mathbb{Z}^+}$ where the child distribution is the same as the marginal distribution of each vertices degree in $\mathcal{G}_1(n, c)$. Let $\rho_n = \rho_n(c)$ be the survival probability of this process, which we know tends to 0 if $c \leq 1$ and a positive value otherwise.

By the "law of rare events", the distribution $\delta_i$ converges to a Poisson($c$) distribution. Before proceeding we prove that this implies that $\rho_n(c) \to \rho(c)$ as $n \to \infty$, where $\rho(c)$ is the survival probability of GW process with Poisson($c$) offspring. This continuity result isn't new[1] and seems to be assumed by some works on random graphs, but since we need it explicitly at several points below, we include it here.

**Lemma 5.4.1.** *Let for each $k \in \mathbb{N}$, $\{Z_i^k\}_{i=0}^{\infty}$ be a Galton-Watson branching process with offspring given by a non-degenerate distribution with probability generating function $f_k$ and extinction probability $q_k$ (so that $q_k = f(q_k)$).*

*If $f_k \to f$ as $n \to \infty$ pointwise, then $q_k \to q$, the smallest value in $[0, 1]$ for which $f(q) = q$.*

*Proof.* Recall that the probability generating functions involved are convex, and take the value 1 at 1. Choose a subsequence $k_i$ such that $q_{k_i} \to \bar{q}$.

$$\bar{q} \leftarrow q_{k_i} = f_{k_i}(q_{k_i}) \to f(\bar{q})$$

all as $n \to \infty$. It follows that $\bar{q}$ is a fixed point of $f$. If $q = 1$ then that is $f$'s only fixpoint in $[0, 1]$, and it follows directly that $\bar{q} = q$ and the result is established. If $q < 1$, $f$ now has two fixpoints in $[0, 1]$ by convexity: 1 and $q$. We must rule out the case $\bar{q} = 1$.

Assume that $q_{k_i} \to 1$. Let $q < s < q_{k_i}$, which implies that $f_{k_i}(s) > s$. Letting $i \to \infty$ for all $q < s < 1$, $f(s) \geq s$. But then there cannot exists a fixpoint $q < 1$ such that $f(q) = q$, which contradicts out assumption. □

**Theorem 5.4.2.** *If $G \sim \mathcal{G}_1(n, c)$ then*

$$\frac{|C_1(G)|}{n} \xrightarrow{p} \rho \quad as\ n \to \infty$$

The proof largely follows the proof for Erdős-Rényi type graphs (see [37], and also [12]. We follow the proof of Lemma 9.6 in the latter without significant deviation up to the proof of Claim 2 in the latter half). The difference is that, because of the clustering, the branching process coupling breaks down sooner. Therefore we need a different argument for why all "large" clusters are in fact the same.

*Proof.* Choose $c'$ and $\epsilon$ such that $1 < c' < c$ and $0 < \epsilon < 1 - 1/c'$.

We construct the graph using the following well known coupling: For every pair of vertices $u$ and $v$, let $U_{u,v}$ be a random variable uniformly distributed on $[0, 1]$. We add an edge between $u$ and $v$ if $U_{u,v} < p_{u,v}$.

Let $G \sim \mathcal{G}_1(n, c')$ constructed in this manner. Later we will increase the $p_{u,v}$ by

$$\frac{\delta}{h_{1,n} d(u, v)} \tag{5.2}$$

where $\delta = c - c'$. The resulting graph is distributed the same as $\mathcal{G}_1(n, c)$.

Note that

$$\log(n) \leq h_{1,n} \leq 2\log(n) \tag{5.3}$$

Consider a standard exploration process on $G$ starting at a vertex $x$. We terminate the exploration either when the explored set becomes larger than some function $\omega(n) \leq n^\epsilon$ where $\omega(n) \to \infty$ as $n \to \infty$ or when the exploration process dies. Following [12] (but with a modified definition) we call such functions *admissible* if for any $\gamma > 0$

$$\omega(n)/n^\gamma \to 0$$

as $n \to \infty$. Let the set $B = B_\omega$ be set of $x$ for which the process stopped for the former reason ($B$ thus contains all the vertices in components larger than $\omega(n)$).

Since $\omega(n) \ll n^\epsilon$, the exploration may be coupled between two branching processes. From above, we can couple it with $Z^n(c')$, and from below by a similar process but where the offspring are given a

[1] Thanks to Peter Jagers for helping me with this proof.

random variable $Y$, the degree of $x$ only counting neighbors more than $\omega(n)$ steps away (since the worst case is that we have already explored the $\omega(n)$ nearest vertices). Using (5.3) this gives

$$\mathbf{E}[Y] \geq c' \left(1 - \frac{\log \omega(n)}{\log n}\right) \geq c'(1-\epsilon).$$

Let $\rho' = \rho'(c')$ be the survival probability of this process. It then follows that,

$$\rho' \leq \mathbf{P}(x \in B) \leq \rho(c') + o(1).$$

By selecting $\epsilon$ sufficiently small, we can make $\rho'$ arbitrarily close to $\rho(c')$ (Lemma 5.4.1), while the coupling still holds for $n$ sufficiently large. Thus $\mathbf{P}(x \in B) \to \rho(c')$ as $n \to \infty$. Addition over all the vertices gives

$$\frac{1}{n}\mathbf{E}|B| \to \rho(c'). \tag{5.4}$$

What remains is to show two things:

1. For all admissible $\omega(n)$, $|B|/n \xrightarrow{p} \rho(c')$ as $n \to \infty$.
2. For some admissible $\omega(n)$, $B$ consists of only one component.

To prove the first claim, we note that the derivation of (5.4) did not depend on the choice of $\omega(n)$, and thus holds for all admissible functions. Given such a function, let $\omega'(n)$ be one strictly larger, and let $B$ and $B'$ be their respective sets of vertices in large components (note that $B' \subset B$).

$$\frac{\mathbf{E}|B \backslash B'|}{n} = \frac{\mathbf{E}|B| - \mathbf{E}|B'|}{n} \to 0. \tag{5.5}$$

It follows that if $|B|/n \xrightarrow{p} \rho(c')$ holds for $B$, it must also hold for $B'$, since

$$\begin{aligned}
\mathbf{P}\left(\left|\frac{|B'|}{n} - \rho\right| > \epsilon_1\right) &= \mathbf{P}\left(\left|\frac{|B| - |B \backslash B'|}{n} - \rho\right| > \epsilon_1\right) \\
&\leq \mathbf{P}\left(\left|\frac{|B|}{n} - \rho\right| > \epsilon_1/2\right) \\
&\quad + \mathbf{P}\left(\frac{|B \backslash B'|}{n} > \epsilon_1/2\right) \to 0
\end{aligned}$$

The second term of the convergence follows by (5.5) and the first moment method.

Now let $\omega(n) \leq \log\log(n)$. We will show the claim for this $\omega(n)$, and use the previous result to establish it for any faster growing $\omega(n)$. We now explore from two vertices $x$ and $y$. Start the exploration from $x$ first. At the end of this, we have found a connected subset $C(x)$ of vertices around $x$. Because both the expected degree of each vertex and the variance is constant, a Chebyshev bound shows that the probability that we should encounter a vertex with more than $\omega(n)$ neighbors is $o(1)$. Thus we can assume that $|C(x)| \leq 2\omega(n)$.

$$\mathbf{P}(y \in C(x)) \leq \frac{2\omega(n)}{\log n} = o(1)$$

since at each step of the exploration, the probability that any vertex which is not $y$ is connected to it is less than $1/2\log n$ by (5.3). Next we explore from $y$ and until we have constructed a $C(y)$. In each step of the exploration, the probability we draw a vertex in $C(x)$ next is bounded from above by $\log\log\log n/\log n$, so

$$\mathbf{P}(C(x) \cap C(y) \neq \emptyset) \leq 2\log\log n \frac{\log\log\log n}{\log n} + o(1) = o(1).$$

It follows that:

$$\rho'\rho' - o(1) \leq P(x, y \in B) \leq \rho(c')\rho(c') + o(1)$$

whence $P(x, y \in B) = \rho^2(c')$ and

$$\frac{1}{n^2}\mathbf{E}[\,|B|^2] \to \rho^2$$

as $n \to \infty$. This means that $\mathbf{Var}(|B|/n) \to 0$ and thus $|B|/n \xrightarrow{p} \rho(c')$. The first claim is thus established.

For the second claim, we will add the additional edges that we withheld in the beginning by increasing the threshold for edge existence by (5.2), and show that this connects all large clusters. We start by letting $\omega(n) = \log^4(n)$ and $B$ be as before. We condition on the graph $G$ constructed with the $c'$ threshold, which we may assume has $|B| \geq (\rho - \epsilon_2)n$ (for $\epsilon_2$ arbitrarily small) by the above. From $G$ we create the graph $G^c$ by completing all the connected clusters of $G$ - note that while this adds edges, it does not change the connectivity properties of the graph. In particular, if adding the additional edges makes $B$ a connected component in $G^c$, it does so also in $G$.

Now select from $B$ as many subsets $K_1, K_2, \dots, K_m$ as possible, such that each $K_i$ is a clique in $G^c$, and each $|K_i| = \log^3(n)$. Since $B$ consists only of connected clusters of size at least $\log^4(n)$ in $G^c$, we can select these so that $m = (\rho - \epsilon_2 - o(1))n/\log^3(n)$.

Consider now the graph $H$, created by taking the $K_i$ as vertices, as connecting $K_i$ and $K_j$ if a new edge is created between any two constituent vertices when the $\delta n$ edges are added. Since for any vertices $x$ and $y$, $d(x,y) \leq n/2$

$$\mathbf{P}(x \text{ and } y \text{ are connected by the new edges}) \geq \frac{\delta}{2n\log n}$$

It follows that the number of connections created between $K_i$ and $K_j$ dominates a random variable $X$ which is $\text{Bin}(\log^6(n), \delta/2n\log n)$ distributed. From a simple second moment estimate, one gets

$$\mathbf{P}(X > 0) \geq \frac{\delta \log^6(n)}{4n\log n} = \left(\frac{\delta \log^2(n)}{4}\right)\frac{\log^3(n)}{n}.$$

It follows that $H$ is dominated by a graph of the form

$$\mathcal{G}\left((1-\epsilon_2 - o(1))\frac{n}{\log^3(n)}, \left(\frac{\delta \log^2(n)}{4}\right)\frac{\log^3(n)}{n}\right)$$

of the standard Erdős-Rényi $\mathcal{G}(n,p)$ family. But the threshold for $\mathcal{G}(n,p)$ being completed connected a.a.s. is $p \gg \log n/n$, which holds here. Thus $H$ is a.a.s. connected, from which it follows that $B$ is a.a.s. connected in the completed graph. This establishes the result. □

## 5.5 Conclusion

The $\alpha$-model spans the spectrum from structure-free random graph models to ordinary percolation. When $\alpha \leq 1$, the connectivity results more or less mirror those of random graphs, whereas for greater values they behave more like percolation.

We note that the cases where $\alpha \leq 1$ are exactly those where $p_{u,v} \to 0$ as $n \to \infty$ for all $u \neq v$. An interesting question is to further explore this territory and see if this property, under some regularity (perhaps monotonicity) requirements, is sufficient for "random graph" type behavior, or if there are cases where this holds, but where the critical value is not one.

## Acknowledgments

Thanks to my advisers Olle Häggström, and Devdatt Dubhashi, as well as Oliver Riordan and Peter Jagers.

# Chapter 6

# Distributed Routing

## 6.1 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 [48]. 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 [20].

Models to explain why graphs develop a small diameter ([62], [11], [60]), 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 [39] 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 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.

### 6.1.1 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 [40] and [61] this is motivated by reason of people's group memberships[1]. Several dynamic processes by which networks can evolve to achieve a similar edge distribution have also been proposed recently, for example, in [16], as well as in forthcoming work by this author [56].

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?

### 6.1.2 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 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

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

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.

### 6.1.3 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. [44] 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 [63] 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.

## 6.2 Kleinberg's Model

Kleinberg's small-world model, like that of Watts and Strogatz [62] 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:

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

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 [40], working on non-geographical models, and there used $q$ (node degree) equal to $\kappa \log^2(n)$ for a constant $\kappa$. In this case it is rather easy to see that $\kappa$ 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 $q = \Theta(\log(n) \log\log(n))$[2]. 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.

## 6.3 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.

[2] 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 that a given node is a dead-end is thus bounded by $(\log n)^{\theta}$. $\theta$ 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.

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.

## 6.4 Statement

Let $V$ be a set of nodes. Let $\phi$ be a function from $V$ onto $G$, a finite (and possibly toric) square lattice in $k$ dimensions[3]. $\phi$ is the configuration of positions assigned ! to the nodes in a base graph $G$. Let $d$ denote graph distance in $G$. Thus for $x, y \in V$, $d(\phi(x), \phi(y))$ 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 $\phi$ and $G$. In particular, if we let $x_j$ and $y_j$ denote the start and end point, respectively, of edge $j$, then:

$$\Pr(E|\phi) = \prod_{i=1}^{m} \frac{1}{d(\phi(x_i), \phi(y_i))^k H_G} \tag{6.1}$$

where $H_G$ is a normalizing constant.

When seen as a function of $\phi$, (6.1) is the likelihood function of a certain configuration having been used to generate the graph. The most

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

straightforward manner in which to estimate $\phi$ from a given realization $E$ is to choose the maximum likelihood estimate, that is the configuration $\hat{\phi}$ which maximizes (6.1). 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 [31], 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 $\phi$ 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:

$$\Pr(\phi|E) = \frac{\Pr(E|\phi)\Pr(\phi)}{\Pr(E)} \tag{6.2}$$

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.

### 6.4.1 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 $\pi$ on a set $S$, to construct a Markov chain on $S$ with $\pi$ 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 $\alpha$ and $\pi$. 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 $\alpha$ is, and can quite easily be shown to have $\pi$ as its stationary distribution, see [35], [36]. The mixing properties of the Markov chain depend on $\alpha$, but beyond that the selection kernel can be chosen as need be.

### 6.4.2 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 (6.2) as its stationary distribution [4]. Let $\alpha$ be a selection kernel on $S$, and $\phi_2$ be chosen by $\alpha$ from $\phi_1$. It follows that, if we let $\alpha(\phi_1, \phi_2) = \alpha(\phi_2, \phi_1)$, and assume a uniform a-priori distribution, then:

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

Let $\phi_2$ be an $x, y$-switch of $\phi_1$ 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:

$$\beta(\phi_1, \phi_2) = \min\left(1, \prod_{i \in \mathrm{E}(x \vee y)} \frac{d(\phi_1(x_i), \phi_1(y_i))^k}{d(\phi_2(x_i), \phi_2(y_i))^k}\right) \tag{6.3}$$

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 $\phi_2$ as the $x, y$-switch of $\phi_1$. This is equivalent to the

[4] 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 $\beta$ coefficient is 1.

kernel:

$$\alpha(\phi_1, \phi_2) = \begin{cases} 1/(n + \binom{n}{2}) & \text{if } x, y\text{-switch} \\ 0 & \text{otherwise.} \end{cases} \tag{6.4}$$

The Markov chain on $S$ with transition matrix

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

with $\alpha$ and $\beta$ given by (6.4) and (6.3) respectively, is thus the Metropolis-Hastings chain with (6.2) 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 6.8 below.

## 6.5 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/`

## 6.6 Experimental Methodology

### 6.6.1 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 timescales.

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], [33]), so this gives the possibility of comparing our results to that.

### 6.6.2 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.

### 6.6.3 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 $n$:

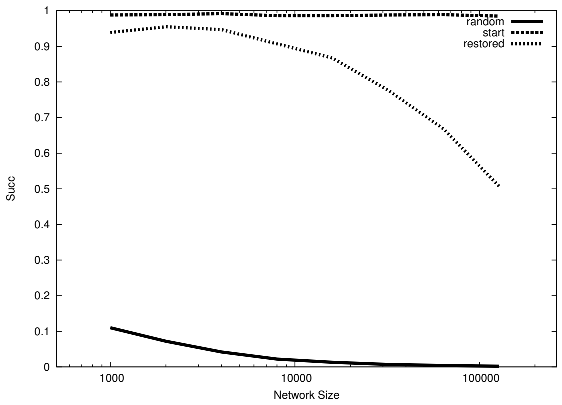


Figure 6.1: The success-rate of queries when terminating at dead-end nodes, on a graph generated by the ideal model.

1. Let $\partial G_n$ 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 $\partial G_n$ 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.

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/`

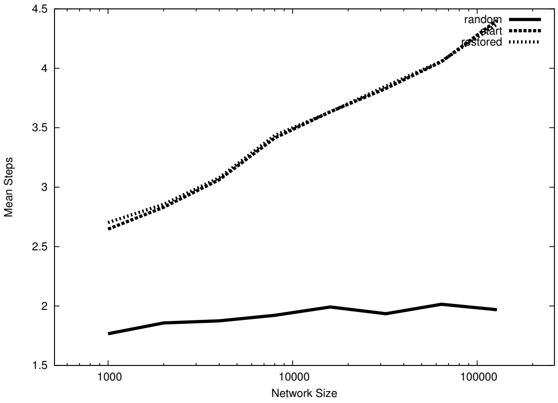


Figure 6.2: Mean number of steps of successful queries when terminating at dead-end nodes, on a graph generated by the ideal model.

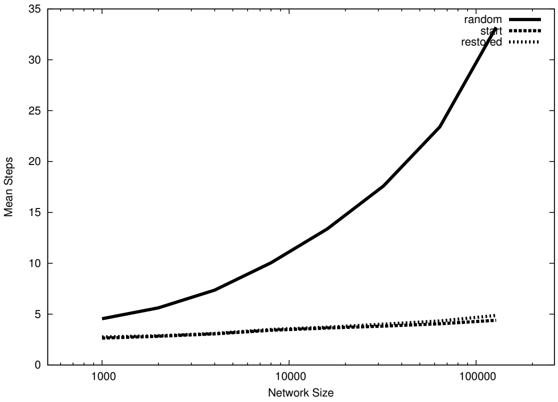


Figure 6.3: Mean number of steps of successful queries when allowed to use local connections, on a graph generated by the ideal model.

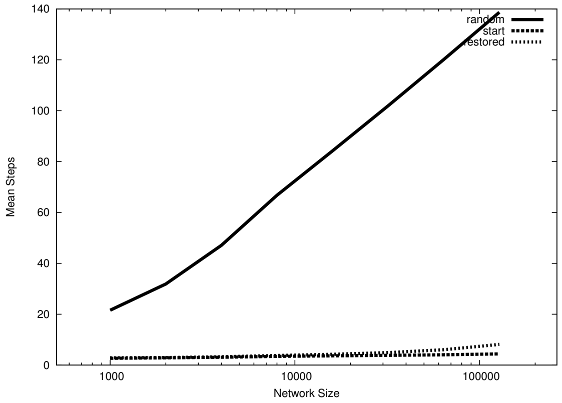


Figure 6.4: Mean number of steps of successful queries when terminating after $(\log_2(n))^2$ steps, on a graph generated by the ideal model.

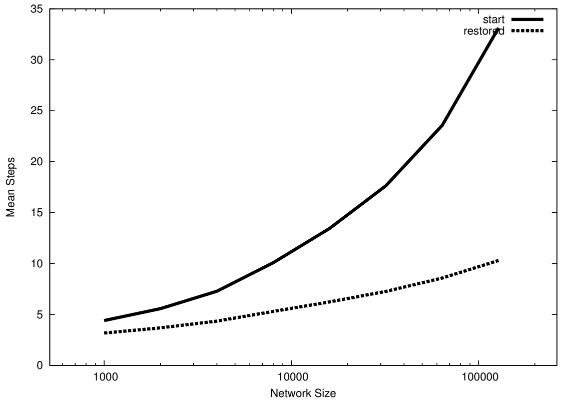


Figure 6.5: Mean number of steps of successful queries when allowed to use local connections, on a graph generated by random matchings.

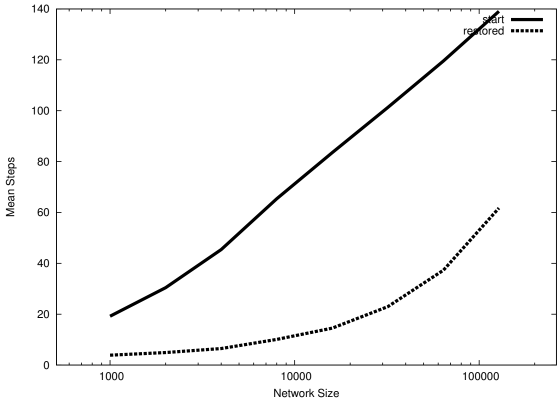


Figure 6.6: Mean number of steps of successful queries when terminating after $(\log_2(n))^2$ steps, on a graph generated by random matchings.

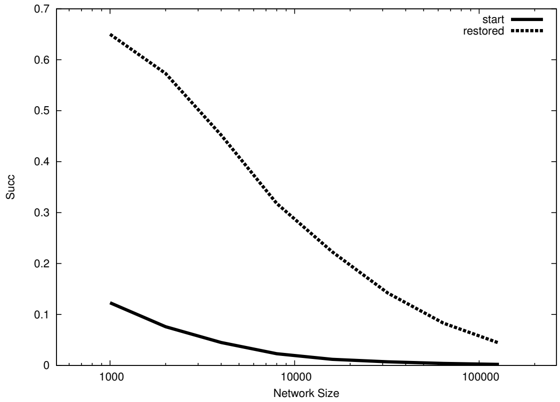


Figure 6.7: The success-rate of queries when terminating at dead-end nodes, on a graph generated by random matchings.

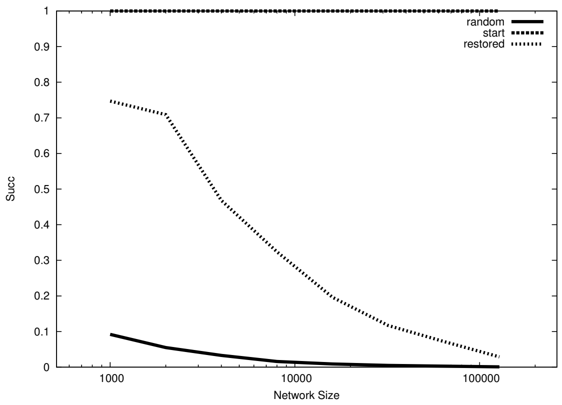


Figure 6.8: The success-rate of queries when terminating at dead-end nodes, on a graph generated with connection probabilities inverse square proportional to the length.

## 6.7 Experimental Results and Analysis

### 6.7.1 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 6.1 through 6.8. 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.

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 6.3, 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 6.1 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 6.6 and 6.5. However, it is 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 6.7).

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 6.8 if particularly depressing in this regard. It should be noted that it has been shown [47] 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.

### 6.7.2 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 dimensional case, but it performs better than against the one dimensional algorithm did on the graphs generated from non-ideal models. See Figures 6.9 to 6.11 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 6.12 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

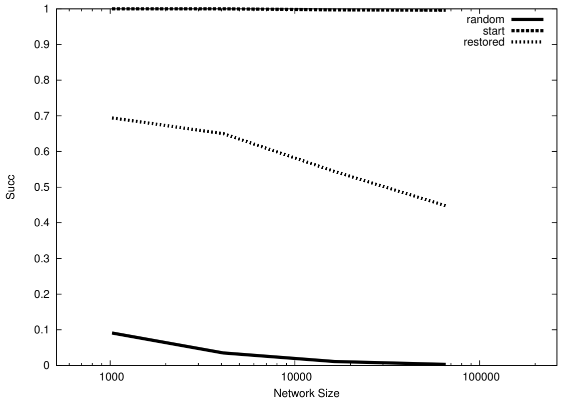


Figure 6.9: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Success rate when failing at dead-end nodes.

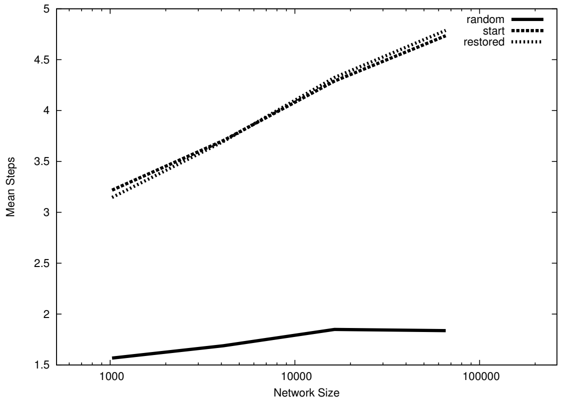


Figure 6.10: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Mean number of steps of successful queries when failing at dead-end nodes.

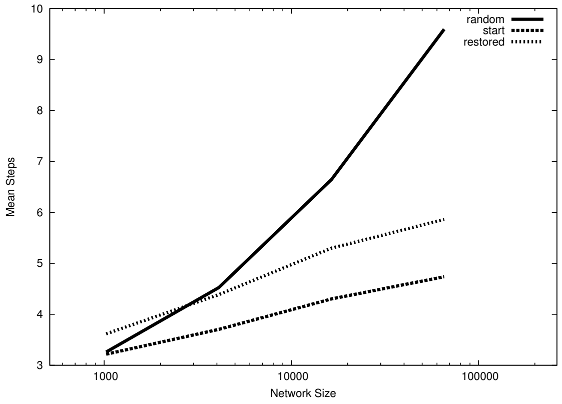


Figure 6.11: Matching Kleinberg's model in 2 dimensions against a graph generated according to it. Mean number of steps of queries when they are allowed to use local connections.

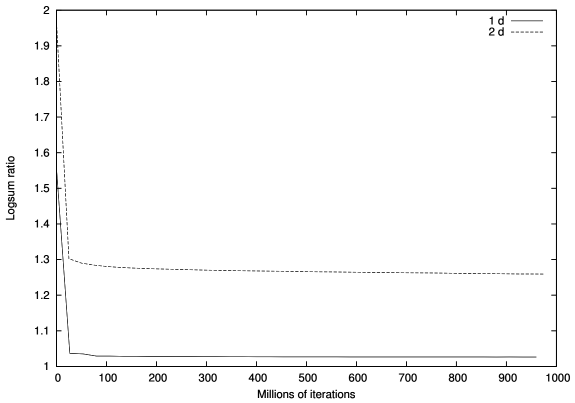


Figure 6.12: The target function of the optimization (log sum of shortcut distances) as the algorithm progresses. The graphs have 10000 nodes with edges generated using the ideal model. The values are normalized by dividing by the log sum of the original graph: it can be seen that we come much closer to restoring this value in 1 dimension.

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.

### 6.7.3 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.[5]

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 [44],

[5] 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 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.

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 [20] 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.

## 6.8 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)[6]. 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 |

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

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 (6.3) 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 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.

## 6.9 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 [39]. His later work on the dynamics of information [40] (and also [61]), 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.

## Acknowledgments

Thanks to my advisers as well as the anonymous reviewers for their input.

# 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] M. Aizenman and C. M. Newman. Discontinuity of the percolation density in one dimensional $1/|x-y|^2$ percolation models. *Communications in Mathematical Physics*, 107:611–647, 1986.

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

[6] 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.

[7] I. Benjamini and N. Berger. The diameter of long-range percolation clusters on finite cycles. *Random Structures and Algorithms*, 19:102–111, 2001.

[8] T. Blass. *The Man Who Shocked the World – The Life and Legacy of Stanley Milgram.* Basic Books, 2004.

[9] B. Bollobás. The diameter of random graphs. *Transactions of the American Mathetical Soceity*, 267:41–52, 1981.

[10] B. Bollobás. *Random Graphs.* Academic Press, 1985.

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

[12] B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structure and Algorithms*, 31:3–122, 2007.

[13] B. Bollobás and O. Riordan. *Handbook of Graphs and Networks*, chapter Mathematical results on scale-free random graphs, pages 1–34. Wiley-VCH, Weinheim, 2003.

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

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

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

[17] D. Coppersmith, D. Gamarik, and M. Sviridenko. The diameter of a one-dimensional long-range percolation graph. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 329 – 337, 2002.

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

[19] M. Dell'Amico. Mapping small worlds. In *Proceedings of the 7th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2007.

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

[21] M. Draief and A. Ganesh. Efficient routing in Poisson small-world networks. *Journal of Applied Probability.*, 43:678–686, 2006.

[22] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Could any graph be turned into a small world? *Theoretical Computer Science*, 355:96 – 103, 2006.

[23] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Towards small world emergence. In *Proceedings of 18th ACM Symposium on Parallelism in Algorithms and Architectures*, 2006.

[24] R. Durrett. *Random Graph Dynamics*. Cambridge University Press, New York, 2005.

[25] D. Eppstein and J. Y. Wang. A steady state model for graph power laws. In *2nd International Workshop on Web Dynamics*, May 2002.

[26] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[27] N. S. Evans, C. GauthierDickey, and C. Grothoff. Routing in the dark: Pitch black. In *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, 2007.

[28] P. Fraigniaud. Greedy routing in tree-decomposed graphs: a new perspective on the small-world phenomenon. In *Proceedings of the 13th European Symposium on Algorithms (ESA)*, pages 791–802, 2005.

[29] P. Fraigniaud, E. Lebhar, and Z. Lotker. A doubling dimension threshold theta(loglog n) for augmented graph navigability. In *Proceedings of the 14th European Symposium on Algorithms (ESA)*, 2006.

[30] M. Franceschetti and R. Meester. Navigation in small world networks, a scale-free continuum model. *Journal of Applied Probability*, 43:1173–1180, 2006.

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

[32] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30:1141–1144, 1959.

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

[34] G. Grimmett. *Percolation.* Springer Verlag, Berlin, 2nd edition, 1999.

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

[36] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

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

[38] J. Kleinberg. Navigation in a small world. *Nature*, page 845, 2000.

[39] J. Kleinberg. The small-world phenomenon: an algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, 2000.

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

[41] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

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

[43] R. Kumar, D. Liben-Nowell, J. Novak, P. Raghavan, and A. Tomkins. Theoretical analysis of geographic routing in social networks. Technical Report MIT-CSAIL-TR-2005-040, Computer Science and Artificial Intelligence Labratory, MIT, 2005.

[44] 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.

[45] G. Singh Manku. Know thy neighbor's neighbor: the power of lookahead in randomized P2P networks. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, 2004.

[46] G. Singh Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in small world. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, 2003.

[47] 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 the principles of distributed computing*, pages 179–188, 2004.

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

[49] O. Mogren. Dynamics of geographical routing in small-world networks. Master's thesis, Chalmers University of Technology, Göteborg, Sweden, June 2007.

[50] C.M. Newman and L.S. Schulman. One dimensional $1/|j-i|^s$ percolation models: The existance of a transition for $s \leq 2$. *Communications in Mathematical Physics*, 104:547–571, 1986.

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

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

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

[54] O. Sandberg. Distributed routing in small-world networks. In *Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX06)*, 2006.

[55] O. Sandberg. Neighbor selection and hitting probability in small-world graphs. To appear in *The Annals of Applied Probability*, 2007.

[56] O. Sandberg and I. Clarke. The evolution of navigable small-world networks. Technical Report 2007:14, Department of Computer Science and Engineering, Chalmers University of Technology, 2007.

[57] R. Solomonoff and Rapoport A. Connectivity of random nets. *Bulletin Mathetical Biophysics*, 13:107–117, 1951.

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

[59] V. Verendel. Switching for a small world. Master's thesis, Chalmers University of Technology, Göteborg, Sweden, June 2007.

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

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

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

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

[64] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve Freenet performance. In *Proceedings of IEEE Infocom*, 2002.