Greedy Navigation in Kleinberg's Small-World Network with Directional Constraints

Chenkai Wang

Abstract

This project explores the small-world phenomenon by first introducing its conceptual background, followed by the construction of a representative network model. The model's properties are rigorously examined through theoretical analysis. In addition, a greedy routing algorithm is implemented to study the relationship between delivery time and the clustering exponent, showing strong agreement between theoretical predictions and simulation results. test test sync.

Keywords: Small-World Network, Greedy Algorithm, Kleinberg Model, Expected Delivery Time

南方科技大学

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Contents

1 Description of Project

When applying Kleinberg's Navigation model on a two-dimensional network, what is the optimal value of α if long-range connections can only be generated along the horizontal and vertical directions?

Requirements:

- Write according to the paper format, covering the title, abstract, and other main parts.
- The project should include both theoretical analysis and network analysis.
- Be clear; both Chinese and English are acceptable.
- Try to illustrate with more graphs.
- Submit on time.

2 Introduction of the problem

2.1 Small world phenomena

In 1967, Milgram completed his famous sociological small world experiment[3], revealing that we are much closer to a stranger than we think. More specifically, a randomly selected pair of individuals can typically be connected through only 5.5 intermediaries on average. Figure 1 illustrates one such possible connection path.

Figure 1: One possible path in the small world experiment

In conclusion, short paths are ubiquitous in the real world. This naturally leads to the question: how can we efficiently discover such short paths? In section 5, we will provide an algorithm to solve it.

2.2 From the reality to the network

Think of every person on the planet as a node. If two people know each other, use a bilateral edge to connect the nodes of those two people so that we can get a social network. A natural hypothesis is that if we could construct such a network, we would be able to determine the distance between any pair of individuals. However, building such an explicit network is impossible. There are at least three reasons:

- Storing a network with such a large number of nodes and edges is computationally expensive;
- people are socializing and making new friends anytime and anywhere, and the network cannot give real-time feedback;
- we do not care about the exact distance of two picked people. Instead, we focus more on the average distance between two people and the existing path.

Therefore, we need to build a network that has the following features. On the one hand, the construction rules should be simple enough to facilitate analysis across different network scales. On the other hand, the simple network structure allows for meaningful interpretation of the small-world phenomenon.

3 The construction of the network

We derive a network below from an $n \times n$ lattice with two kinds of edges to meet the requirement.

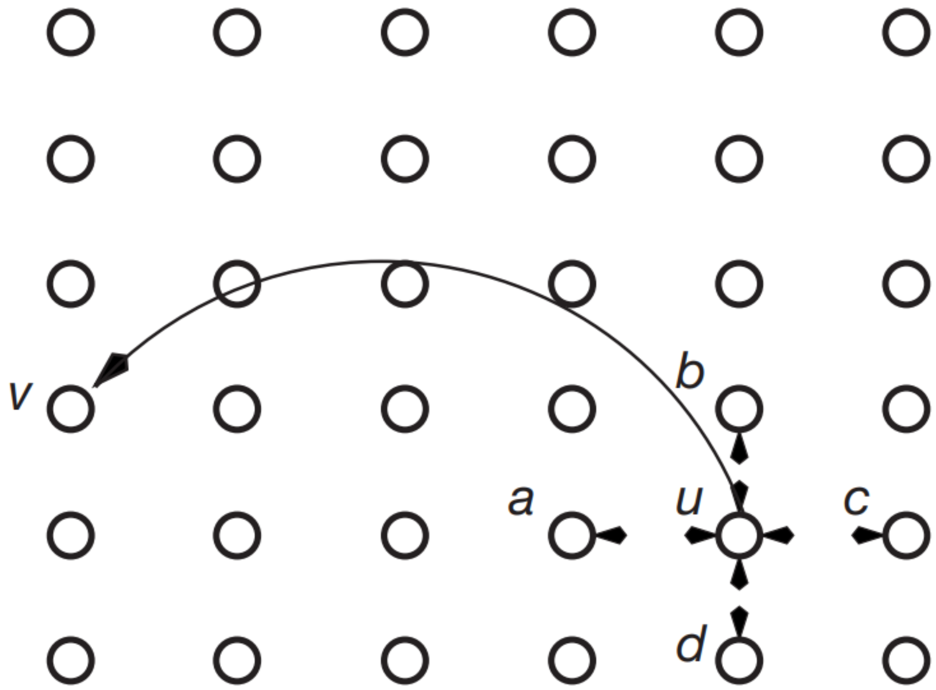


Figure 2: The built network[2]

- The former is the directed short-range edge. Each node, say u, has a short-range connection to its nearest neighbors, say a, b, c, d.
- The latter we call it directed long-range edge. For each node, say u, has a probability proportional to $d(u,v)^{-r}$ to be connected to a randomly chosen node v, where r is the fixed clustering coefficient and $d(u,v)$ denotes the Manhattan distance of u and v.

For long-range edges, we will add a constraint that edges can only be made in the horizontal or vertical direction and compare it to the case with no constraint. The comparison part will be demonstrated in the next section.

To continue, we need to provide a more precise definition. In the above $n \times n$ network $\{(i,j) : i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,n\}\}$, the distance between any two nodes $u(i,j)$ and $v(k,l)$ is defined as $d(u,v) = |k-i|+|l-j|$. Moreover, we introduce two universal parameters, p, and q. The node u has a directed short-range edge to its nearest neighbors with distance p. In figure 2, $p = 1$. Besides that, each node u has q directed long-range edges. In figure 2, $q = 1$.

4 The proof and properties for the built network

4.1 The proof of two important theorems

In Section 3, it was discussed that there are two different networks - one with a constraint and one without. This section aims to demonstrate the properties of the network with the constraint, using theorems and proofs primarily borrowed from the work of Kleinberg in 2000[1].

Theorem 1 (original theorem 1). *(a) Let $0 \leq r < 2$. There is a constant α_r, depending on p, q, r, but independent of n, so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(2-r)/3}$.*

(b) Let $r > 2$. There is a constant α_r, depending on p, q, r, but independent of n, so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(r-2)/(r-1)}$.

We have a similar result for the network with the constraint, denoted as adjusted theorem 1.

Theorem 2 (adjusted theorem 1). *(a) Let $0 \leq r < 1$. There is a constant α_r, depending on p, q, r, but independent of n, so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(1-r)/3}$.*

(b) Let $r > 1$. There is a constant α_r, depending on p, q, r, but independent of n, so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(r-1)/r}$.

Proof. Here is the proof of the theorem 2(a).

The probability that a node u chooses v as its i^{th} out of q long-range contacts is $d(u,v)^{-r} / \sum_{v \neq u} d(u,v)^{-r}$, and we have

$$\begin{aligned}
\sum_{v \neq u} d(u,v)^{-r} &\geq \sum_{j=1}^{n/2} (1)\left(j^{-r}\right) = \sum_{j=1}^{n/2} j^{-r} \\
&\geq \int_{1}^{n/2} x^{-r} dx \\
&\geq (1-r)^{-1}\left((n/2)^{1-r} - 1\right) \\
&\geq \frac{1}{(1-r)2^{2-r}} \cdot n^{1-r}
\end{aligned}$$

where the last line follows if we assume $n \geq 2^{2-r}$. Let $\delta = (1-r)/3$.

Let U denote the set of nodes within lattice distance pn^{δ} of t. Note that

$$|U| \leq 1 + \sum_{j=1}^{pn^{\delta}} 4 \leq 4p^2n^{2\delta}$$

where we assume n is large enough that $pn^{\delta} \geq 2$. Define $\lambda = 1/\left((1-r)2^{6-r}qp^2\right)$. Let $\mathcal{E}'$ be the event that within λn^{δ} steps, the message reaches a node other than t with a long-range contact in U. Let $\mathcal{E}'_i$ be the event that in step i, the message reaches a node other than t with a long-range contact in U; thus $\mathcal{E}' = \bigcup_{i \leq \lambda n^{\delta}} \mathcal{E}'_i$. Now, the node reached at step i has q long-range contacts that are generated at random when it is encountered, so we have

$$\begin{aligned}\Pr\left[\mathcal{E}'_i\right] &\leq \frac{q|U|}{\frac{1}{(1-r)2^{2-r}} \cdot n^{1-r}} \\ &= \frac{(1-r)2^{4-r}qp^2n^{2\delta}}{n^{1-r}}\end{aligned}$$

Since the probability of a union of events is bounded by the sum of their probabilities, we have

$$\begin{aligned}\Pr\left[\mathcal{E}'\right] &\leq \sum_{i \leq \lambda n^{\delta}} \Pr\left[\mathcal{E}'_i\right] \\ &\leq \frac{(1-r)2^{4-r}qp^2n^{2\delta}}{n^{1-r}} \\ &= (21-r)2^{4-r}\lambda qp^2 \leq \frac{1}{4}\end{aligned}$$

We now define two further events. Let $\mathcal{F}$ denote the event that the chosen source s and the target t are separated by a lattice distance of at least $n/4$. One can verify that $Pr\left[\mathcal{F}\right] \geq \frac{1}{2}$. Since $Pr\left[\overline{\mathcal{F}} \vee \mathcal{E}'\right] \leq \frac{1}{2} + \frac{1}{4}$, $\Pr\left[\mathcal{F} \wedge \overline{\mathcal{E}'}\right] \geq \frac{1}{4}$.

Finally, Let X denote the random variable equal to the number of steps taken for the message to reach t, and let $\mathcal{E}$ denote the event that the message reaches t within λn^{δ} steps. We claim that if $\mathcal{F}$ occurs and $\mathcal{E}'$ does not occur, then $\mathcal{E}$ cannot occur. For suppose it does. Since $d(s,t) \geq n/4 \geq p\lambda n^{\delta}$, in any $s-t$ path of at most λn^{δ} steps, the message must be passed at least once from a node to a long-range contact. Moreover, the final time this happens, the long-range contact must lie in U. This contradicts our assumption that $\mathcal{E}'$ does not occur.

Thus $Pr\left[\mathcal{E} \mid \mathcal{F} \wedge \overline{\mathcal{E}'}\right] = 0$, hence $E\left[X \mid \mathcal{F} \wedge \overline{\mathcal{E}'}\right] \geq \lambda n^{\delta}$. Since

$$E\left[X\right] \geq E\left[X \mid \mathcal{F} \wedge \overline{\mathcal{E}'}\right] \cdot \Pr\left[\mathcal{F} \wedge \overline{\mathcal{E}'}\right] \geq \frac{1}{4}\lambda n^{\delta}$$

Let $\alpha_r = \frac{1}{4}\lambda$, since $\delta = \frac{1-r}{3}$, we know that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(1-r)/3}$, completing the proof. □

Proof. Here is the proof of the theorem 2(b).

Consider a node u, and let v be a randomly generated long-range contact of v. For any m, we have

$$\begin{aligned}\Pr[d(u,v) > m] &\leq \sum_{j=m+1}^{2n-2} (4)\left(j^{-r}\right)\\ &= 4\sum_{j=m+1}^{2n-2} j^{-r}\\ &\leq \int_{m}^{\infty} x^{1-r}dx\\ &\leq (r-1)^{-1}m^{1-r} = \varepsilon^{-1}m^{-\varepsilon}\end{aligned}$$

where $\varepsilon = r - 1$.

We set $\beta = \frac{\varepsilon}{1+\varepsilon}, \gamma = \frac{1}{1+\varepsilon}$, and $\lambda' = \frac{min(\varepsilon,1)}{8q}$. We assume n has been chosen large enough that $n^{\gamma} \geq p$. Similar to part (a), we have

$$E\left[X\right] \geq E\left[X \mid \mathcal{F} \wedge \overline{\mathcal{E}'}\right] \cdot \Pr\left[\mathcal{F} \wedge \overline{\mathcal{E}'}\right] \geq \frac{1}{4}\lambda' n^{\beta}$$

Let $\alpha_r = \frac{1}{4}\lambda'$, since $\beta = \frac{\varepsilon}{1+\varepsilon} = \frac{r-1}{r}$ the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(r-1)/r}$, completing the proof. □

Theorem 3 (original theorem 2)**.** *There is a decentralized algorithm $\mathcal{A}$ and a constant α_2, independent of n, so that when $r = 2$ and $p = q = 1$, the expected delivery time of $\mathcal{A}$ is at most $\alpha_2(logn)^2$.*

We have a similar result for the network with the constraint, denoted as adjusted theorem 2.

Theorem 4 (adjusted theorem 2)**.** *There is a decentralized algorithm $\mathcal{A}$ and a constant α_2', independent of n, so that when $r = 1$ and $p = q = 1$, the expected delivery time of $\mathcal{A}$ is at most $\alpha_2'(logn)^2$.*

Proof. Here is the proof of theorem 4. The probability that u chooses v as its long-range contact is $d(u,v)^{-1}/\sum_{v\neq u} d(u,v)^{-1}$ and we have

$$\begin{aligned}\sum_{v\neq u} d(u,v)^{-1} &\leq \sum_{j=1}^{2n-2} (4)\left(j^{-1}\right)\\ &\leq 4 + 4\ln(2n-2)\\ &\leq 4\ln(6n)\end{aligned}$$

Thus, the probability that v is chosen is at least $d(u,v)^{-1}/(4ln(6n))$. Let B_j be the set of nodes within lattice distance 2^j of t. There are at least

$$1 + 4 \times \sum_{i=1}^{2^j} 1 > 2^{j+2}$$

nodes in B_j, each is within lattice distance $2^{j+1} + 2^j < 2^{j+2}$ of u. If any of these nodes is the long-range contact of u, it will be $u's$ closest neighbor to t; thus, the message enters B_j with the probability at least

$$\frac{2^{j+2}}{4\ln(6n)2^{j+2}} = \frac{1}{4\ln(6n)}$$

Let X_j denote the total number of steps spent in phase j, $log(logn) \leq j < logn$. We have

$$E\left[X_j\right] = \sum_{i=1}^{\infty} Pr\left[X_j \geq i\right] = 4ln(6n)$$

Let X denote the total number of steps spent by the algorithm. We have

$$X = \sum_{j=0}^{\log n} X_j$$

and so by the linearity of expectation we have $E\left[X_j\right] \leq (1+logn)(4ln(6n)) \leq \alpha_2'(logn)^2$ for a suitable choice of α_2', completing the proof. □

4.2 The properties of two networks based on the theorems

In this section, we will provide a summary of two network properties, which are listed as follows.

- For the unconstrained network, we have

$$E[X] \geq \alpha_r n^{(2-r)/3}, \quad 0 \leq r < 2$$
$$E[X] \geq \alpha_r n^{(r-2)/(r-1)}, \quad r > 2$$

  optimal $r = 2$.

- For the constrained network, we have

$$E[X] \geq \alpha_{r'} n^{(1-r')/3}, \quad 0 \leq r' < 1$$
$$E[X] \geq \alpha_{r'} n^{(r'-1)/(r')}, \quad r' > 2$$

  optimal $r' = 1$.

- The relation between r and r' is $r' = r - 1$. which reveals that there exists a translation when adding the constraint.

- The expected delivery time of the decentralized algorithm $\mathcal{A}$ with $p = q = 1$ and optimal $r's$ for both cases are at most $constant \times (logn)^2$, where different networks correspond to different constants.

5 The greedy algorithm and its simulation

5.1 The greedy algorithm

To efficiently find a path from a source node u to a target node v, we employ a greedy navigation algorithm. The algorithm iteratively selects the neighbor of the current node that is closest to v, considering both short-range and long-range connections. The following presents the algorithm:

Algorithm 1 Greedy Navigation Algorithm

Input: Grid size n, parameters p, q, clustering exponent r, source node u, target node v

Output: Navigation path S from u to v

1: Initialize path $S \leftarrow \{u\}$
2: **while** $u \neq v$ **do**
3: $\quad S_{\text{temp}} \leftarrow \emptyset$
4: $\quad$ **// Add short-range neighbors within distance** p
5: $\quad$ **for** each node w such that $1 \leq d(u, w) \leq p$ **do**
6: $\quad\quad$ Add w to S_{temp}
7: $\quad$ **end for**
8: $\quad$ **// Add** q **long-range neighbors sampled by** $d(u, z)^{-r}$**, constrained to axis-aligned directions**
9: $\quad$ **for** $i = 1$ to q **do**
10: $\quad\quad$ Sample distance d with probability $\propto d^{-r}$
11: $\quad\quad$ Uniformly choose $(\Delta x, \Delta y)$ such that $|\Delta x| + |\Delta y| = d$ and $(\Delta x = 0) \vee (\Delta y = 0)$
12: $\quad\quad$ $z \leftarrow$ node at $(u_x + \Delta x, u_y + \Delta y)$ with toroidal wrapping
13: $\quad\quad$ Add z to S_{temp}
14: $\quad$ **end for**
15: $\quad \omega \leftarrow \arg\min_{x \in S_{\text{temp}}} d(x, v)$
16: $\quad$ Append ω to S
17: $\quad u \leftarrow \omega$
18: **end while**

The input parameters have already been explained before. Notice that u always gets closer to v after each circulation, the algorithm is guaranteed to converge since each step reduces the distance to the target. We know that the greedy algorithm obtains an optimal local solution, but the locally optimal solution is not always the optimal global solution. However, here we do not care whether we get global optimal or not due to the following reasons:

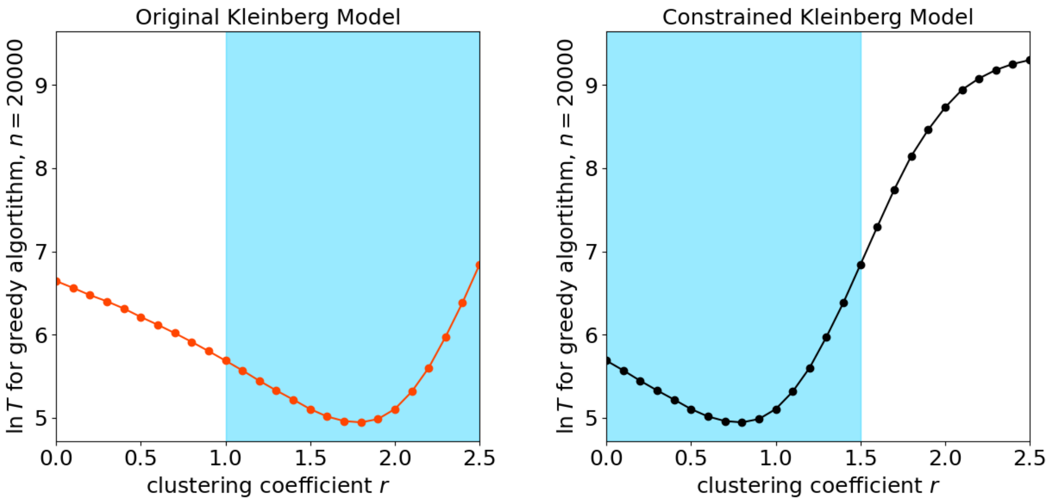
- it is almost impossible to get a global optimal;
- even if we could get a globally optimal, it is not affordable;
- we get local optimal with very little time and space complexity compared to global optimal, and we are happy to lose a little accuracy in exchange for a huge increase in efficiency.

The greedy algorithm can be applied to both network types by following the appropriate constraints in steps 4 and 5.

5.2 Simulation of the greedy algorithm

The primary figure of merit is its expected cost time T, which represents the expected number of steps from the specified node u to the target node v. In contrast to algorithms requiring global knowledge of the network, the greedy algorithm finds reasonably short paths using only local information.

Now we set $n = 20000$, $u = (1, 1)$, $v = (20000, 20000)$, $p = 1$, $q = 1$. We want to find the relationship between expected delivery time T and clustering coefficient r. Let r increase from 0 by 0.1 to 2.5 and record the delivery time T of each r. We can get the following pictures for both networks.



(a) without constraint (b) with constraint

Figure 3: The relationship between expected delivery with and without constraint

It is observed in Figure 3a and Figure 3b that the minimum delivery time is attained at $r = 2$ in the absence of constraints and at $r = 1$ in the presence of constraints. Furthermore, it is noted that the blue-shaded area remains consistent across both scenarios, which is in concordance with the results outlined in Section 4.2.

6 Conclusion

Our findings in sections 3 through 5 demonstrate strong agreement between theoretical analysis and empirical simulation. We observe that when there is no constraint, the delivery time reaches its minimum at $r = 2$, and when there is a constraint, the minimum occurs at $r = 1$. Additionally, the directional constraint introduces a shift in the optimal exponent, effectively translating the theoretical results, as discussed in Section 4.2.

References

[1] Jon Kleinberg. "The small-world phenomenon: An algorithmic perspective". In: *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. 2000, pp. 163–170.

[2] Jon M Kleinberg. "Navigation in a small world". In: *Nature* 406.6798 (2000), pp. 845–845.

[3] Stanley Milgram. "The small world problem". In: *Psychology today* 2.1 (1967), pp. 60–67.