The Small-World Phenomenon

The world is small in the sense that "short" paths exists between almost any two people.

How short are these paths?

How can we measure their length?

Milgram Small World Experiment

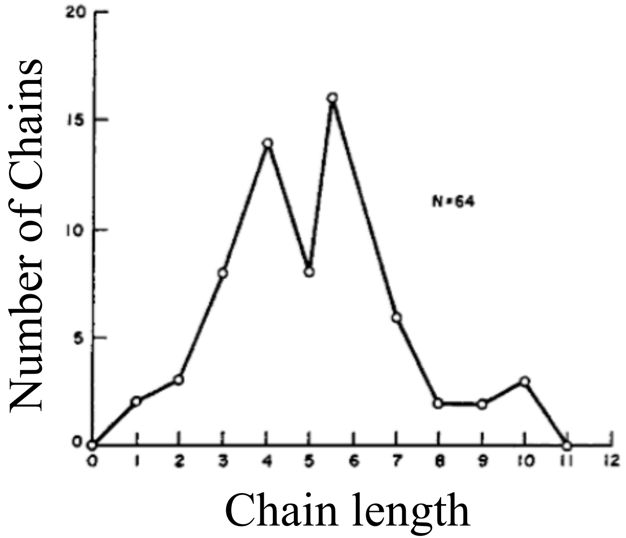
Set up (1960s):

- 296 randomly chosen "starters" asked to forward a letter to a "target" person.
- Target was a stockbroker in Boston.
- Instructions for starter:
  - Send letter to target if you know him on a first name basis.
  - If you do not know target, send letter (and instructions) to someone you know on a first name basis who is more likely to know the target.
- Some information about the target, such as city, and occupation, was provided.

Milgram Small World Experiment

Results:

- 64 out of the 296 letters reached the target.
- Median chain length was 6 (consistent with the phrase "six degrees of separation")

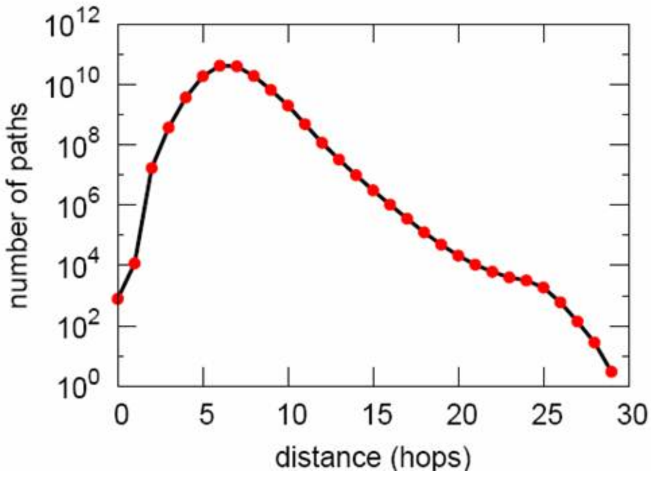


Key points:

- A relatively large percentage (>20%) of letters reached target.
- Paths were relatively short.
- People were able to find these short paths.

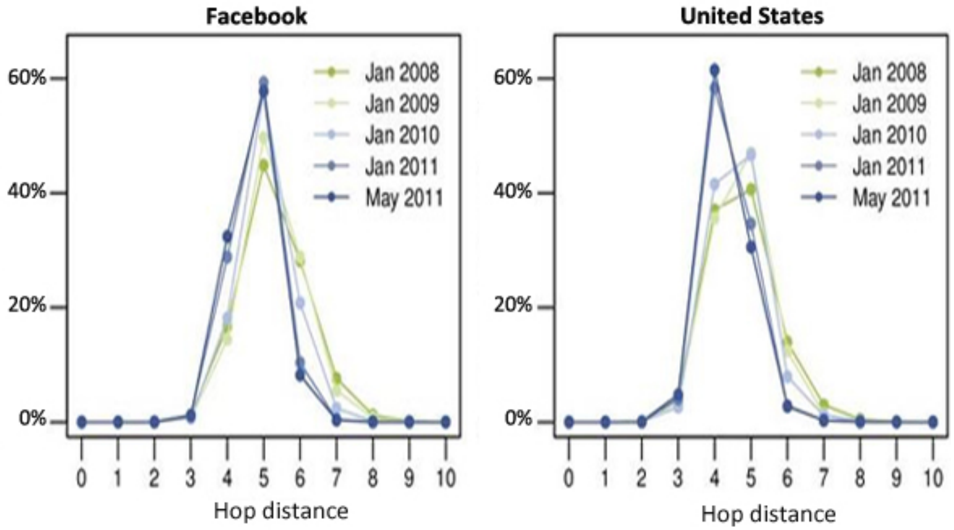
Small World of Instant Message

- Nodes: 240 million active users on Microsoft Instant Messenger.
- Edges: Users engaged in two-way communication over a one-month period.
- Estimated median path length of 7.



[Leskovec and Horvitz, 2008]

Small World of Facebook



[Backstrom et al. 2012]

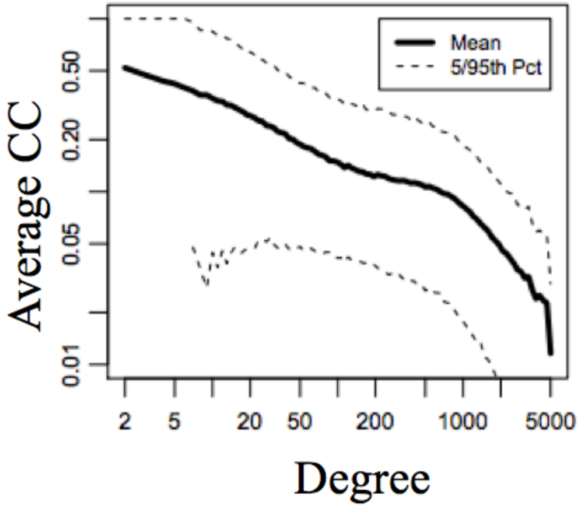
- Global network: average path length in 2008 was 5.28 and in 2011 it was 4.74.
- Path are even shorter if network is restricted to US only.

Clustering Coefficient

Local clustering coefficient of a node:
Fraction of pairs of the node's friends that are friends with each other.

- Facebook 2011: High average CC (decreases with degree)
- Microsoft Instant Message: Average CC of 0.13.
- IMDB actor network: Average CC 0.78

In a random graph, the average clustering coefficient would be much smaller.



[Ugander et al. 2012]

Path Length and Clustering

Social networks tend to have high clustering coefficient and small average path length.

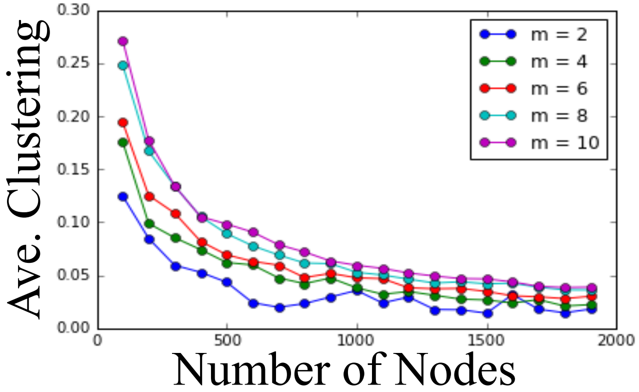
Can we think of a network generative model that has these two properties?

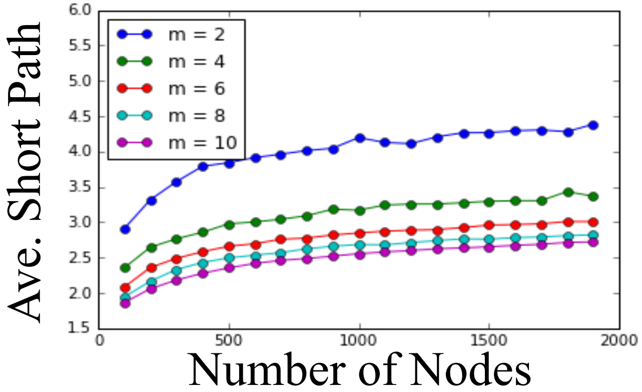
How about the Preferential Attachment model?

```
In: G = nx.barabasi_albert_graph(1000,4)

In: print (nx.average_clustering(G))
Out: 0.0202859273671

In: print (nx.average_shortest_path_length(G))
Out: 4.16942942943
```

Path Length and Clustering

What if we vary the number of nodes (n) or the number of edges per new node (m)?





Small average shortest path: high degree nodes act as hubs and connect many pairs of nodes.

For a fixed m, clustering coefficient becomes very small as the number of nodes increases.

No mechanism in the Preferential Attachment model favors triangle formation.

Small World Model

Motivation: Real networks exhibit high clustering coefficient and small average shortest paths. Can we think of a model that achieves both of these properties?

Small-world model:

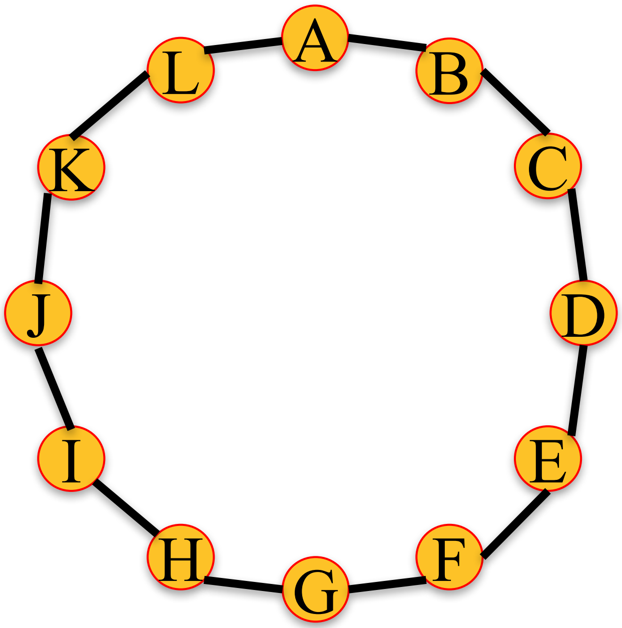
- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$



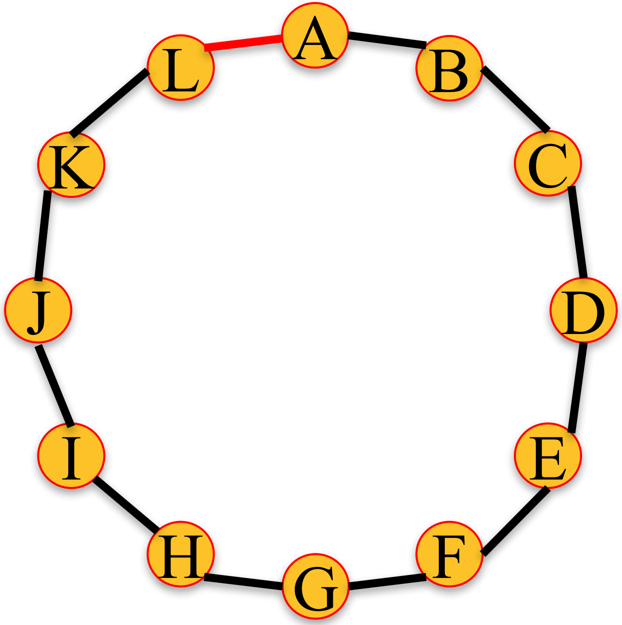
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



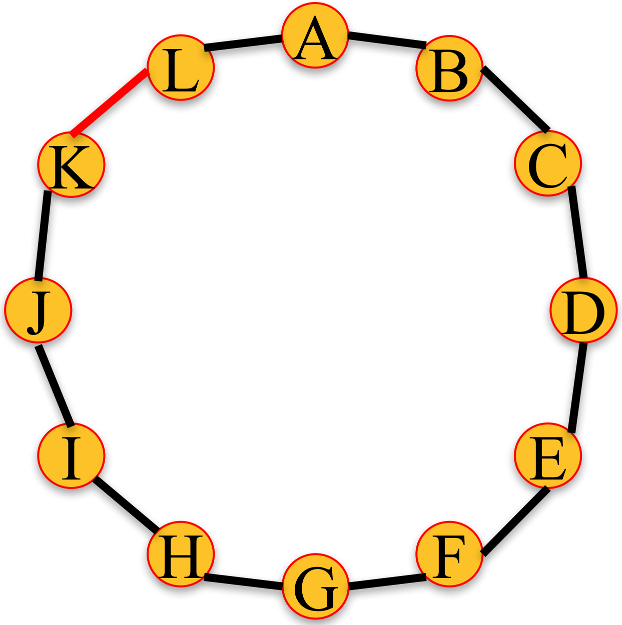
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



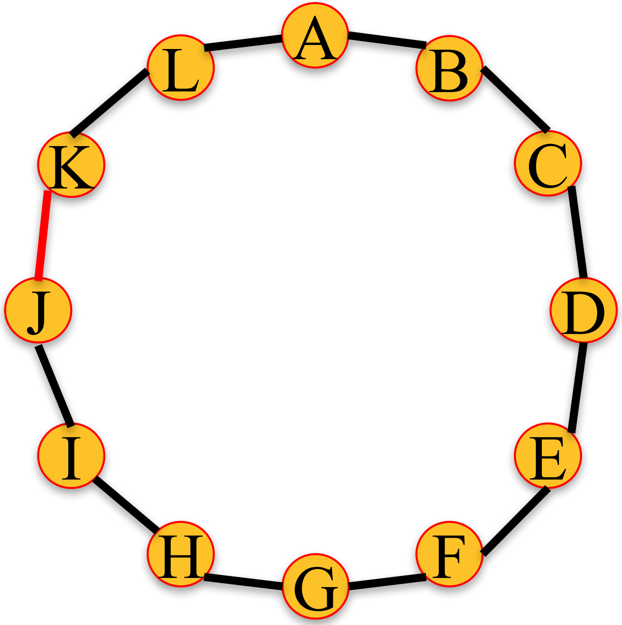
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



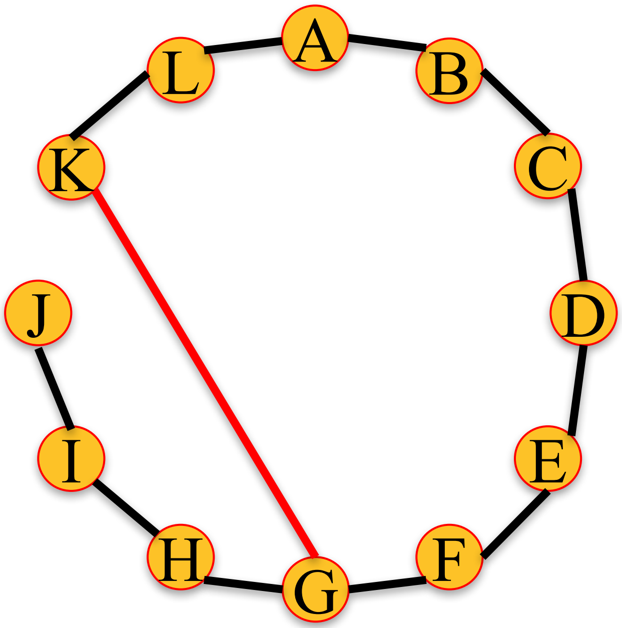
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



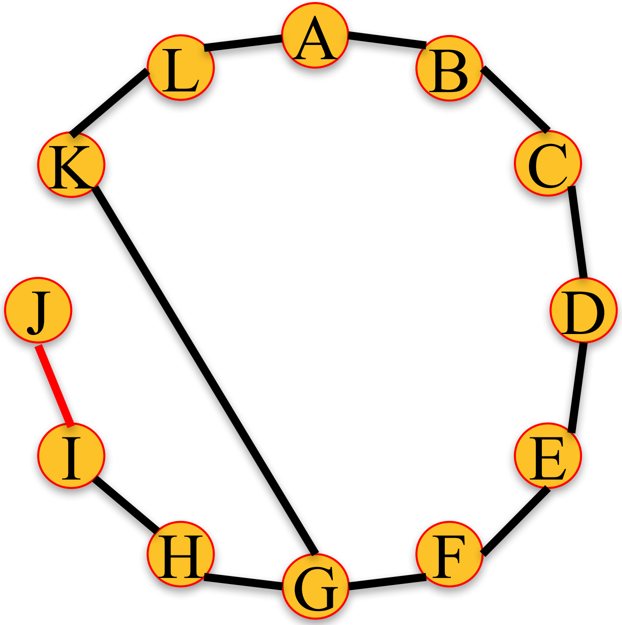
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



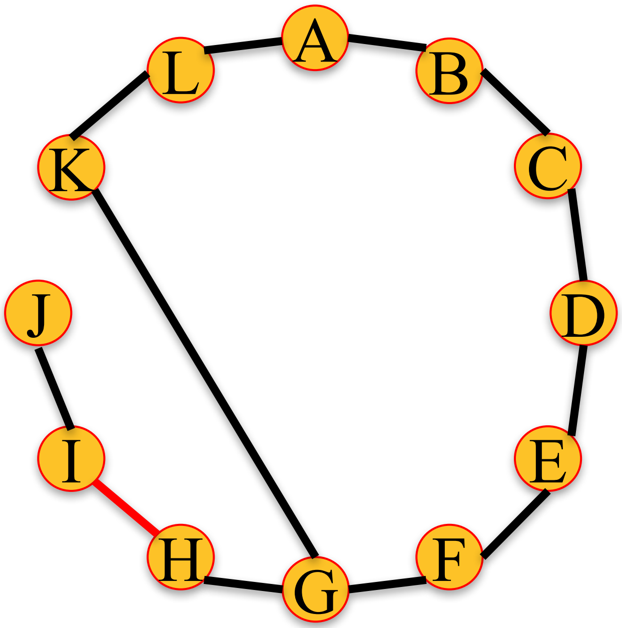
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



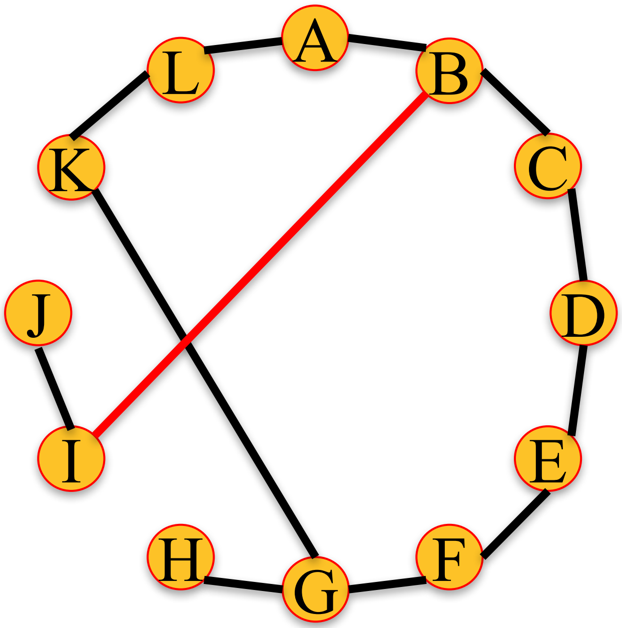
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



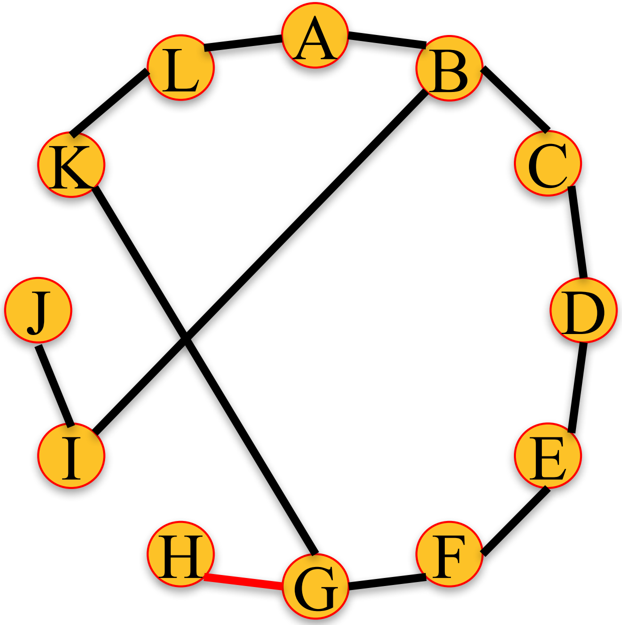
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



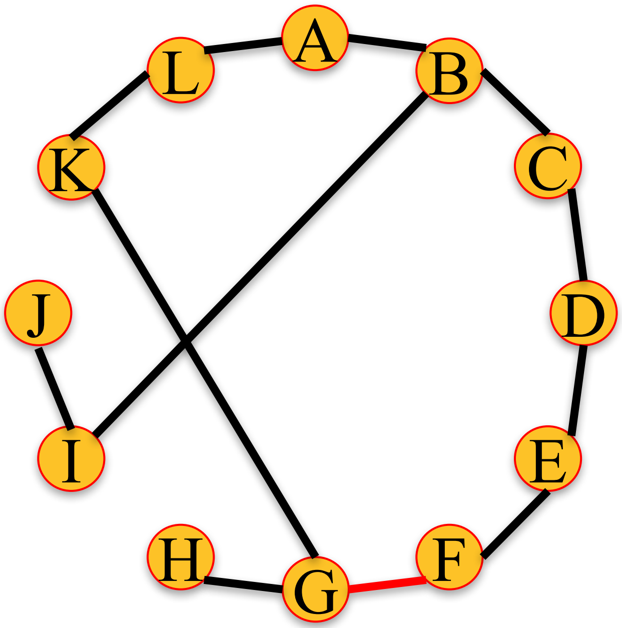
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



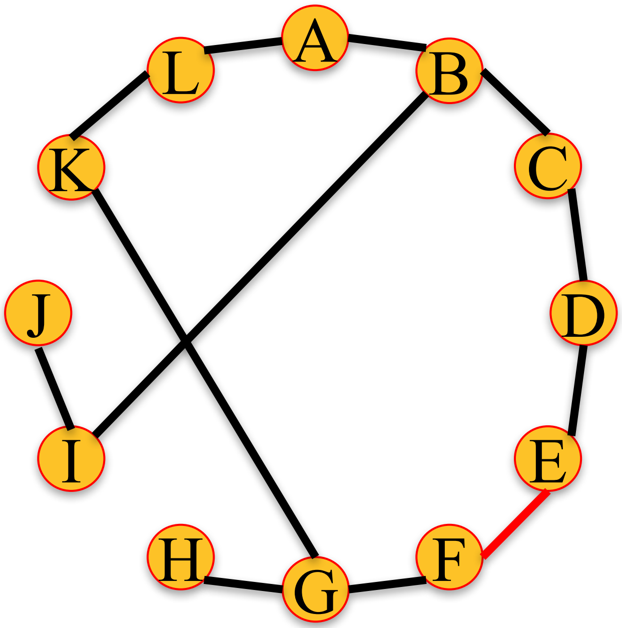
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



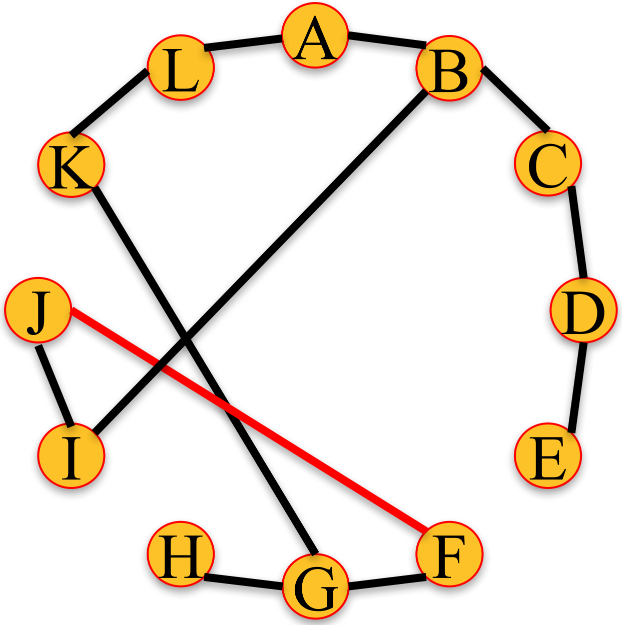
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



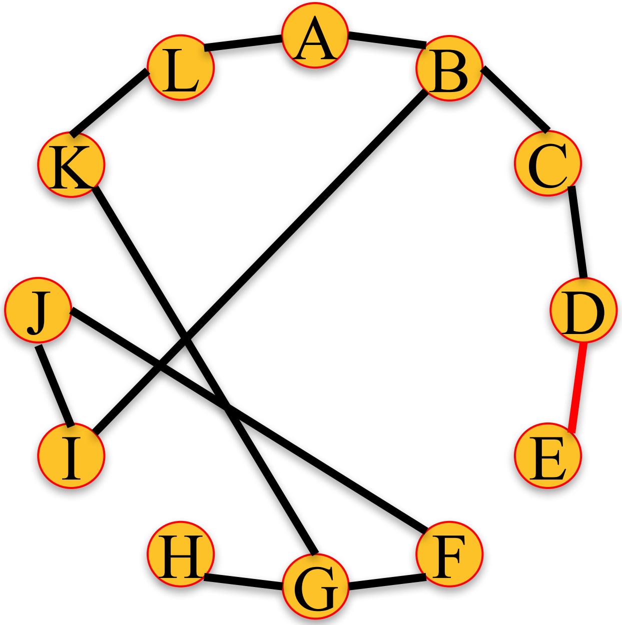
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!



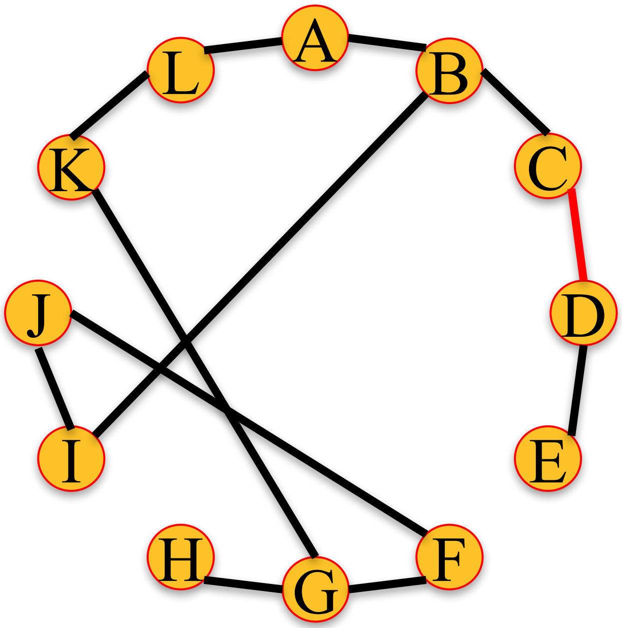
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



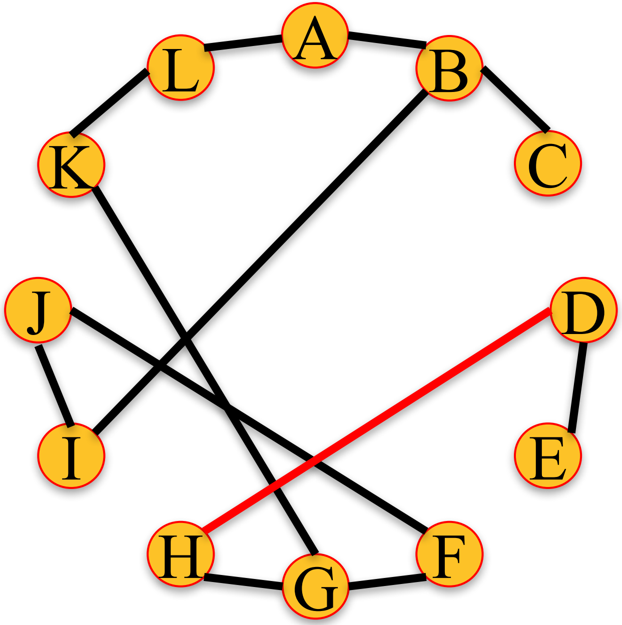
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



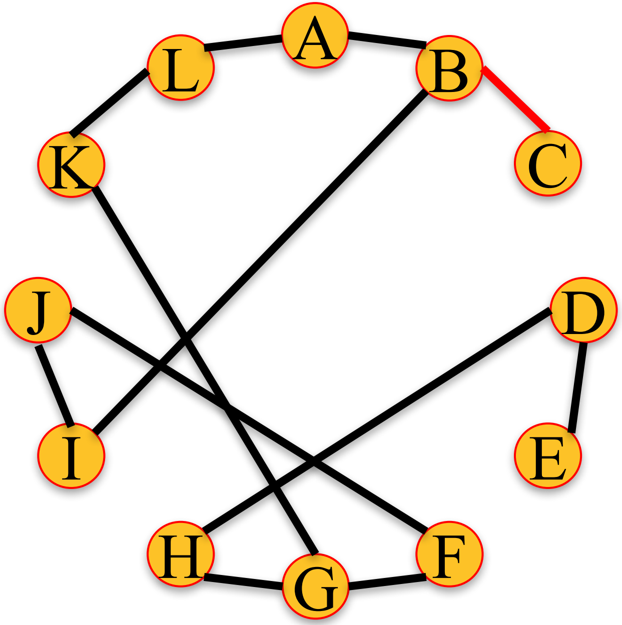
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



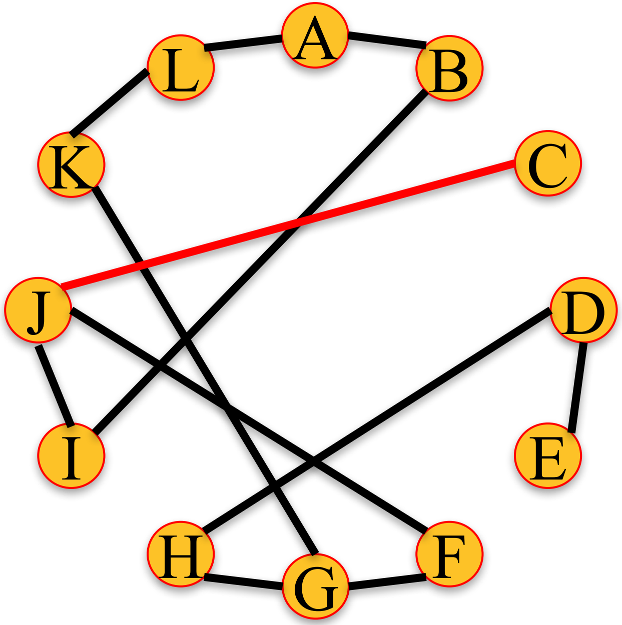
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

Rewire!



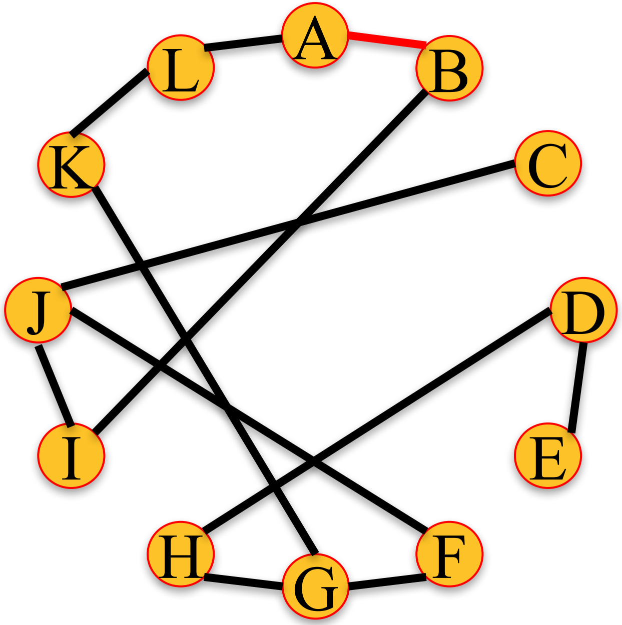
Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

No rewire!

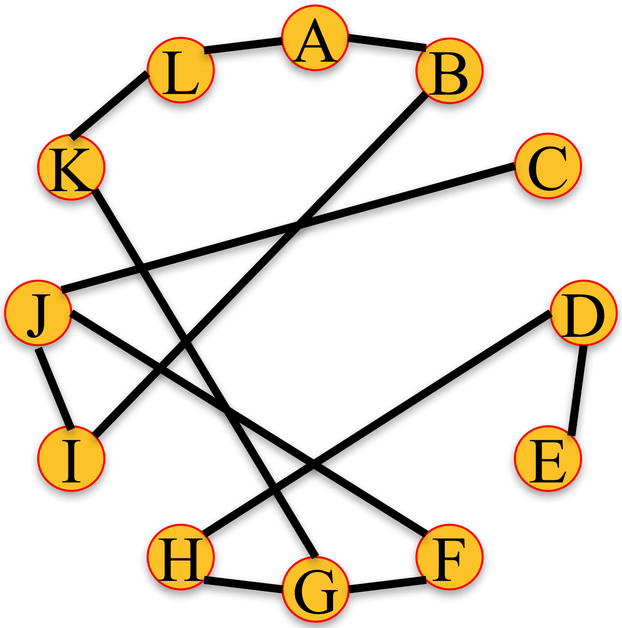


Small World Model

Small-world model:

- Start with a ring of n nodes, where each node is connected to its k nearest neighbors.
- Fix a parameter $p \in [0,1]$
- Consider each edge (u, v). With probability p, select a node w at random and rewire the edge (u, v) so it becomes (u, w).

Example: $k = 2, p = 0.4$

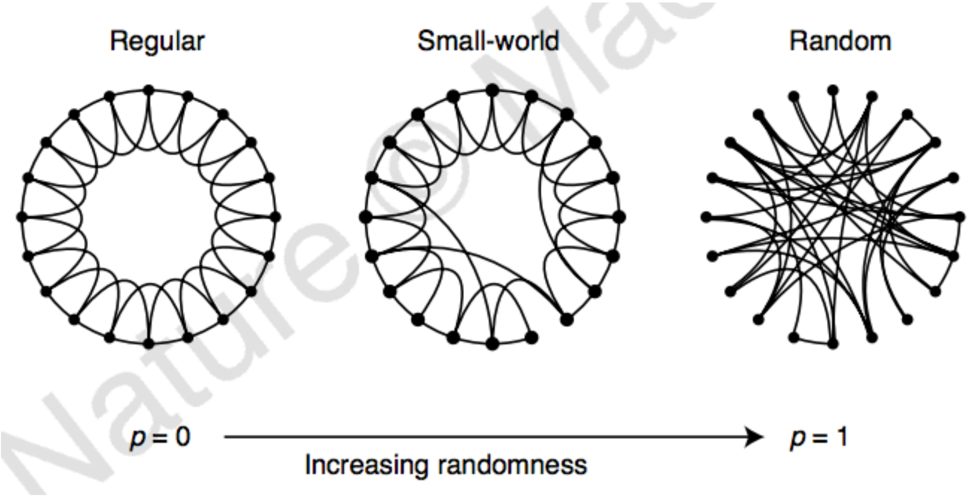


Small World Model

Regular Lattice ($p = 0$): no edge is rewired.

Random Network ($p = 1$): all edges are rewired.

Small World Network ($0 < p < 1$): Some edges are rewired. Network conserves some local structure but has some randomness.



Watts and Strogatz, 1999

Small World Model

What is the average clustering coefficient and shortest path of a small world network?

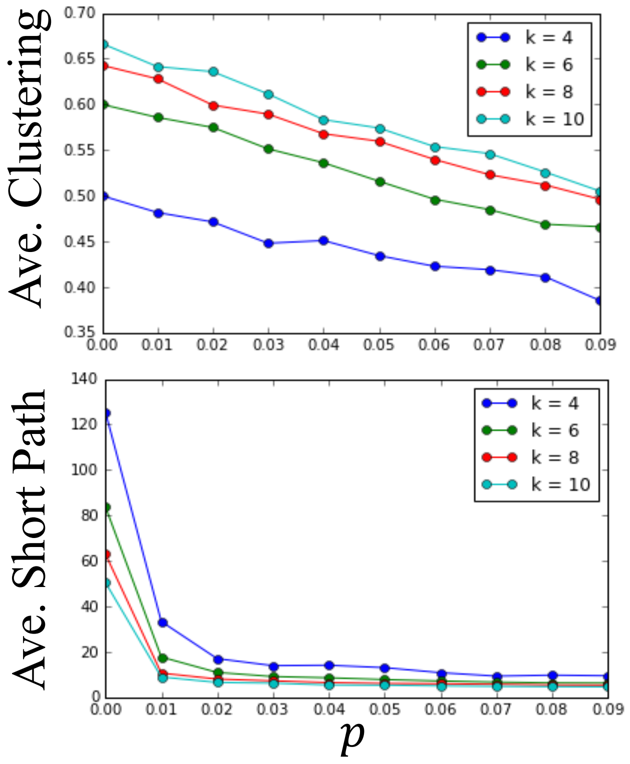
It depends on parameters k and p.

As p increases from 0 to 0.01:

- average shortest path decreases rapidly.
- average clustering coefficient deceases slowly.

An instance of a network of 1000 nodes, $k = 6$, and $p = 0.04$ has:

- 8.99 average shortest path.
- 0.53 average clustering coefficient.

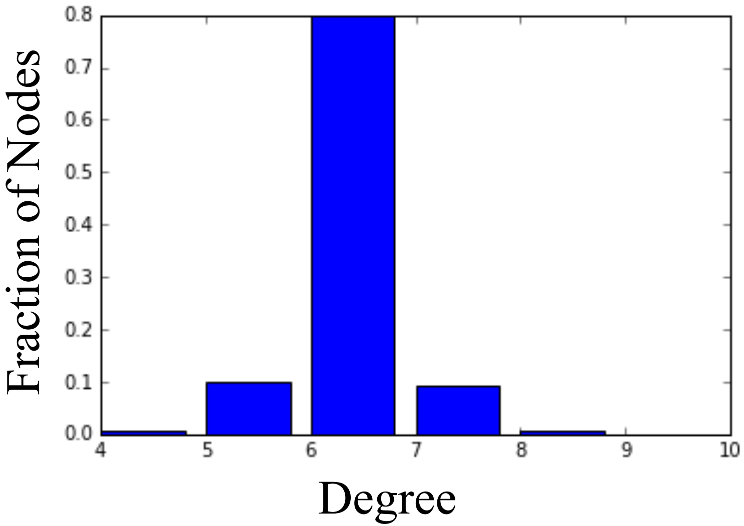


Small World Model in NetworkX

watts_strogatz_graph(n, k, p) returns a small world network with *n* nodes, starting with a ring lattice with each node connected to its k nearest neighbors, and rewiring probability p.

Small world network degree distribution:

```
G = nx.watts_strogatz_graph(1000,6,0.04)
degrees = G.degree()
degree_values = sorted(set(degrees.values()))
histogram =
[list(degrees.values()).count(i)/float(nx.number_of_node
s(G)) for i in degree_values]
plt.bar(degree_values,histogram)
plt.xlabel('Degree')
plt.ylabel('Fraction of Nodes')
plt.show()
```



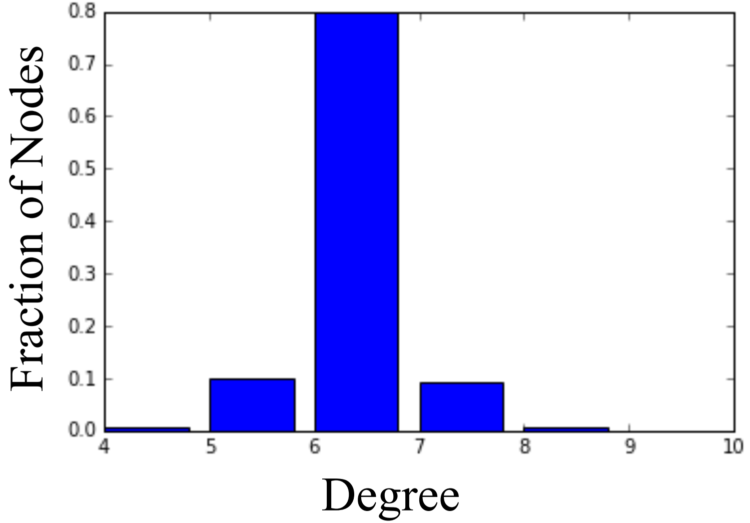
Small World Model in NetworkX

Small world network: 1000 nodes, $k = 6$, and $p = 0.04$

No power law degree distribution.

Since most edges are not rewired, most nodes have degree of 6.

Since edges are rewired uniformly at random, no node accumulated very high degree, like in the preferential attachment model



Small World Model in NetworkX

Variants of the small world model in NetworkX:

- Small world networks can be disconnected, which is sometime undesirable.

connected_watts_strogatz_graph(n, k, p, t) runs *watts_strogatz_graph(n, k, p)* up to *t* times, until it returns a connected small world network.

- *newman_watts_strogatz_graph(n, k, p)* runs a model similar to the small world model, but rather than rewiring edges, new edges are added with probability p.

Summary

- Real social networks appear to have small shortest paths between nodes and high clustering coefficient.
- The preferential attachment model produces networks with small shortest paths but very small clustering coefficient.
- The small world model starts with a ring lattice with nodes connected to k nearest neighbors (high local clustering), and it rewires edges with probability p.
- For small values of p, small world networks have small average shortest path and high clustering coefficient, matching what we observe in real networks.
- However, the degree distribution of small world networks is not a power law.
- On NetworkX, you can use *watts_strogatz_graph(n, k, p)* (and other variants) to produce small world networks.