Watts-Strogatz Small World Simulation

COSC 5302 Advanced Operating System

Project Report

Spring 2015

Table of Contents

List of Tables

List of Figures

1. Project Requirements and Use Cases

This project is to simulate a small-world graph according to Watts-Strogatz Model. The small-world graph consists of a ring lattice with 5000 nodes (N = 5000) and each node has 20 short range neighbors (k = 20). The goal is to understand the properties of small-world graphs through simulation. In this project, average path lengths, clustering coefficient of small-world graphs are computed and analyzed. Through this project, we gain thorough understanding of the Watts-Strogatz model, which explores the formation of networks that result in the "small world" phenomenon.

2. Background and Principles

The Watts-Strogatz model is a random graph generation model that produces graphs with small-world properties, including high clustering coefficient and short average path lengths (Wikipedia 2015b). This model was proposed by Duncan J. Watts and Steven Strogatz in their paper “Collective dynamics of ‘small-world’ networks” published on Nature in 1998 (Watts and Strogatz 1998).

2.1 Clustering Coefficient and Path Length of Graphs

Two main characteristics of small-world graphs are high clustering coefficient and short average path lengths. Clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together (Wikipedia 2015a). The clustering coefficient of a node v in a graph can be defined as the percentage of number of pairs of adjacent neighbors in the total number of pairs of neighbors (Small_World_Networks 2015):

$$cc(v) = \frac{number\ of\ \ pairs\ of\ adjacent\ neighbors}{number\ of\ \ pairs\ of\ neighbors}$$

The clustering coefficient of a graph G is the average of the clustering coefficients of its vertices (Small_World_Networks 2015):

$$CC(G) = \frac{1}{|V|} \sum_{v \in V} cc(v)$$

The rang of the clustering coefficient of a graph CC(G) is between 0 and 1 ($0 \le CC(G) \le 1$). If CC(G) of a graph is close to 1 ($CC(G) \approx 1$), the graph is highly clustered on average. If CC(G) is close to 0, the graph is not highly clustered on average.

Path Length, also known as shortest distance, between two vertices u and v in a graph G is the length of the shortest path from u to v, denoted as d(u, v). The Average Path Length of a connected graph G is the average over pairs of vertices of the path length (Small_World_Networks 2015):

$$L(G) = \frac{1}{n(n-1)} \sum_{u,v} d(u,v)$$

where n is the total number of nodes in the graph. The Average Path Length tells us, on average, the number of steps it takes to get from one node of the graph to another.

2.2 GRAPHS WITH DIFFERENT RANDOMNESS

Figure 2.1 shows three kinds of graphs with different randomness: regular graph, small-world graph and random graph. This figure is cited from Duncan J. Watts and Steven Strogatz's paper "Collective dynamics of 'small-world' networks" (Watts and Strogatz 1998).

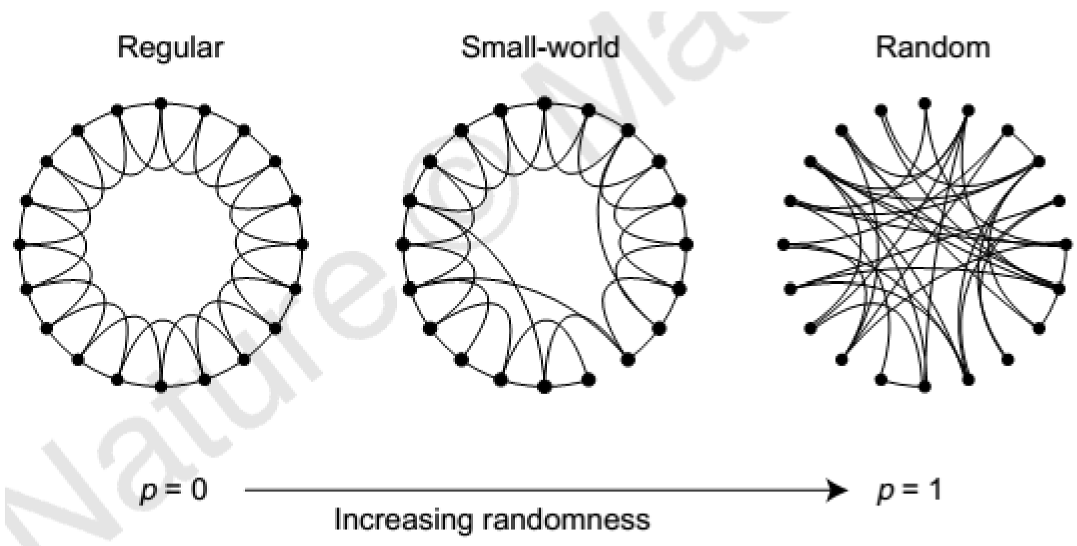

Figure 2.1: Graphs with different randomness. (From: (Watts and Strogatz 1998))

In a regular graph, every node has the same degree, which means every node has the same number of adjacent nodes. In the regular graph example in Figure 2.1, the total number of nodes is 20, and every node connects to its 4 nearest neighbors. The small-world graph and random graph can be created from the regular graph: with probability p, rewiring each edge. When using a probability p of 1, the regular graph will be transformed into a random graph. Some graphs stand in between regular graph and random graph will exhibit properties of a "small world" phenomenon, with high clustering coefficient and short average path lengths. Typically, a regular graph has a high clustering coefficient, but its average path length is long. A random graph has short path length, but the clustering coefficient is small. Interestingly, Small-world graph has high clustering coefficient, like regular graphs, yet has small average path length, like random graphs. Many “real world” biological, technological and social networks fall into Watts-Strogatz small-world model (Watts and Strogatz 1998). Because small-world graphs model lots of natural phenomena, and they have advantages of high clustering coefficient and short average path lengths, scientists and engineers are interested in them.

This project is to simulate and analyze the Watts-Strogatz small-world model.

3. Project Design Choices

The Java programming language is chose to simulate the Watts-Strogatz small-world. As a lot of varying data structures are needed to simulate the system, the advantage of choosing java is that it provides reach libraries, which is convenient for us to implement the system. To accomplish the simulation, the system is designed to consist of five main parts: Graph, GraphDrawer, GraphProcessor, PathExplorer and a client WattsStrogatzSmallWorld.

The Graph is to implement a graph abstraction data type. This is used represent a graph for processing by the GraphProcessor. The API of the Graph is designed as Table 3.1.

Table 3.1: Graph API

public class Graph		
	Graph()	*create an empty graph*
void	addEdge(Integer v, Integer w)	*add edge v-w*
void	deleteEdge(Integer v, Integer w)	*delete edge v-w*
Iterable<Integer>	getNodes()	*get all nodes in the graph*
Iterable<Integer>	adjacentTo(Integer v)	*neighbors of v*
boolean	hasNode(Integer v)	*is v a node in the graph?*
boolean	hasEdge(Integer v, Integer w)	*is v-w an edge in the graph?*

The GraphDrawer is to draw a graph and display it on a window. This is serve to facilitate us observing and analyzing the transformation process of a graph, from regular to random. The API of the GraphDrawer is designed as Table 3.2.

Table 3.2: GraphDrawer API

public class GraphDrawer		
	GraphDrawer(Graph G)	*create a graphDrawer for a graph G*
void	drawGraph(Graph G)	*draw graph G and display*

The GraphProcessor is to process a graph and compute its clustering coefficient and average path length. The API of the GraphProcessor is designed as Table 3.3.

Table 3.3: GraphProcessor API

public class GraphProcessor		
Graph	createRegularGraph(int n, int k)	*create a regular graph*
void	reconnectGraph(Graph G, double p, int k)	*reconnect graph G*
double	avgPathLen(Graph G)	*compute average path lengh*
double	clusterCoeff(Graph G)	*compute lustering coefficient*

The PathExplorer is to find the length of the shortest path from a source node to any other nodes in a graph. The API of the PathExplorer is designed as Table 3.4.

Table 3.4: PathExplorer API

public class PathExplorer		
	PathExplorer(Graph G, Integer s)	*create a path explorer for source s*
boolean	hasPathTo(Integer v)	*Check if v is reachable from source s*
double	pathLenTo(Integer v)	*Get the shortest path length from s to v*

4. Implementation Challenges

One implementation challenge is implementing the Graph abstract data type. A simple way to represent a graph is to maintain a list of the edges in the graph using a linked list or an array. But this implementation is too inefficient. It needs to store a lot of duplicate information. Another representation is the adjacency-matrix representation. This approach needs to maintain a two-dimensional boolean array, which consumes too much space. We use a Map of sets to represent a graph. The key of the Map is a node's ID and the value is its set of neighbors. This representation is widely used and suitable for sparse graphs. As real world graphs tend to be sparse, so this representation is suitable for our project. Using this representation, we can efficiently iterate through all the nodes and their neighbors.

The second implementation challenge is the algorithm used to search the path length. Computing on graphs tend to be time consuming. Efficient algorithm is important for computing graphs. The classic breadth-first search algorithm provides a direct and efficient solution for our project. By applying this algorithm, we are able to compute the path length from a source to a destination in linearithmic time.

The third implementation challenge lies in implementing GraphProcessor. To correctly simulate the Watts-Strogatz small-world model, reconnecting is important and is a key manipulation to transform a regular graph into a small-world graph. The algorithm we use to reconnecting is: For each node i and each edge (i, j) with i < j; with probability p, replace (i, j) with (i, k) where k is chosen uniformly from vertices not equal to or adjacent to i (Small_World_Networks 2015). This implementation satisfies the description of Duncan J. Watts and Steven Strogatz in their paper "Collective dynamics of 'small-world' networks".

5. Results

Figure 5.1 ~ Figure 5.4 show the results of simulation starting with a ring of 100 vertices, each connected to its 4 nearest neighbors by undirected edges. Figure 5.1 is a regular graph, it is the result when reconnecting possibility is 0. We see that the clustering coefficient for a regular graph with n = 100, k = 4, p = 0 (where n is number of nodes, k is number of neighbors, p is reconnecting probability) is 0.5, and the average path length is 12.88, which means the shortest distance between 2 randomly chosen nodes in the graph is 12.88. This graph has a relatively high clustering coefficient, but the average path length is large. These are typical features of a regular graph.

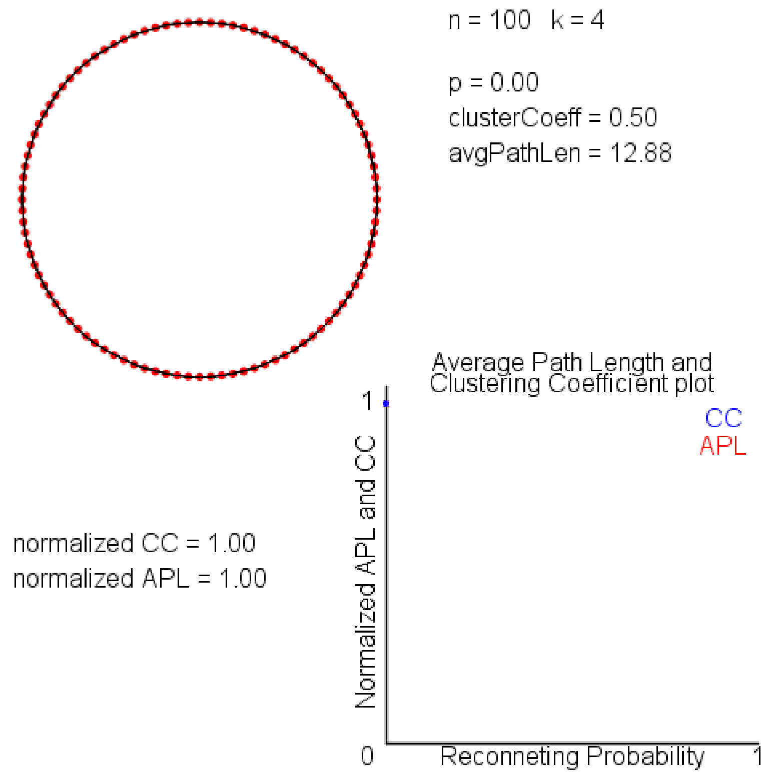

Figure 5.1: Regular graph simulation with p = 0 for n = 100, k = 4.

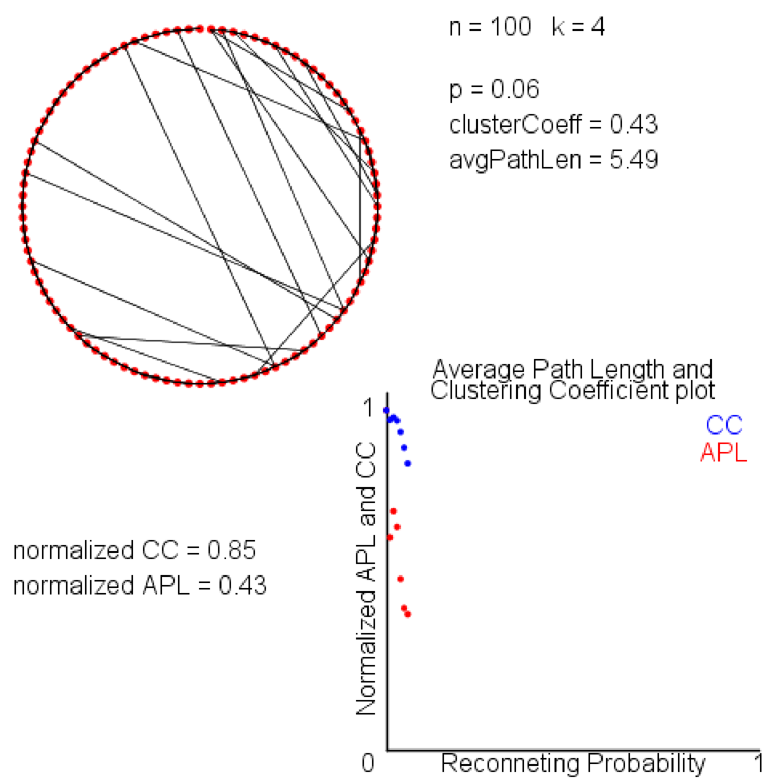

Figure 5.2: Random reconnecting simulation with p = 0.06 for n = 100, k = 4.

Figure 5.2 is the result of simulation for random reconnecting with Reconnecting Probability of 0.06 from a regular graph with n = 100, k = 4. This graph has a clustering coefficient of 0.43, and an average path length of 5.49. The normalized clustering coefficient is 0.85 and normalized average path length is 0.43, which means that comparing with the regular graph in Figure 5.1, clustering coefficient decreases to 85% of original value and average path length decreases to 43% of original value. These results indicate that a little disturber (reconnecting with probability of 0.06) to a regular graph will results in a much larger decrease in average path length than in clustering coefficient. This graph has a relatively high clustering coefficient and short average path length. These are features of a small-world graph.

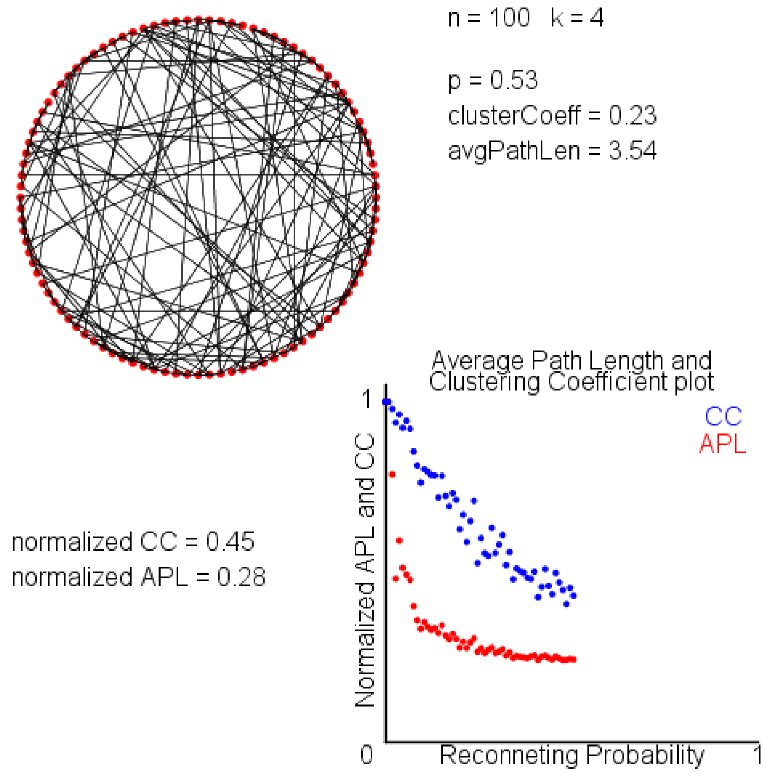

Figure 5.3: Random reconnecting simulation with p = 0.53 for n = 100, k = 4.

Figure 5.3 is the result of simulation for random reconnecting with Reconnecting Probability of 0.53 from a regular graph with n = 100, k = 4. This graph has a clustering coefficient of 0.23, and an average path length of 3.54. Comparing with the regular graph in Figure 5.1, clustering coefficient decreases to 45% of original value and average path length decreases to 28% of original value. The

Average Path Length (APL) and Clustering Coefficient (CC) plot shows that APL decreases much faster than CC.

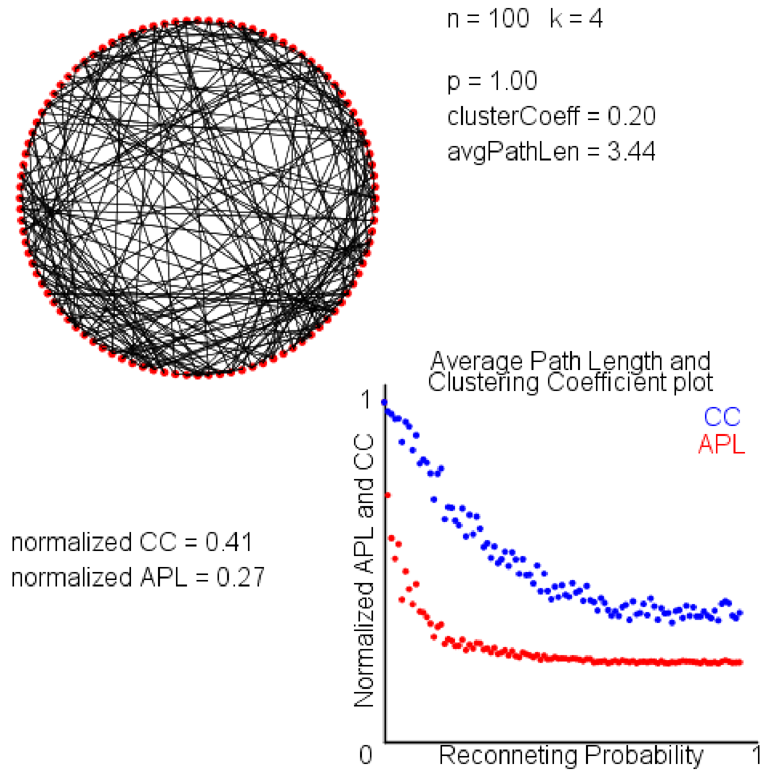

Figure 5.4: Random reconnecting simulation with p = 1.00 for n = 100, k = 4.

Figure 5.4 is the result of simulation for random reconnecting with Reconnecting Probability of 1.00 from a regular graph with n = 100, k = 4. With a Reconnecting Probability of 1.00, the regular graph in Figure 5.1 has been transformed to a random graph. Each node randomly connects to other nodes. This graph has a clustering coefficient of 0.20, and an average path length of 3.44. Comparing with the regular graph in Figure 5.1, clustering coefficient decreases to 41% of original value and average path length decreases to 27% of original value. The Average Path Length (APL) and Clustering Coefficient (CC) plot shows that APL and CC first decrease very fast when Reconnecting Probability is in the range of 0 ~ 0.3, and APL decreases much faster than CC in this range, then they stay stable. This graph has a short average path length, but its clustering coefficient is small. These are typical features of a random graph.

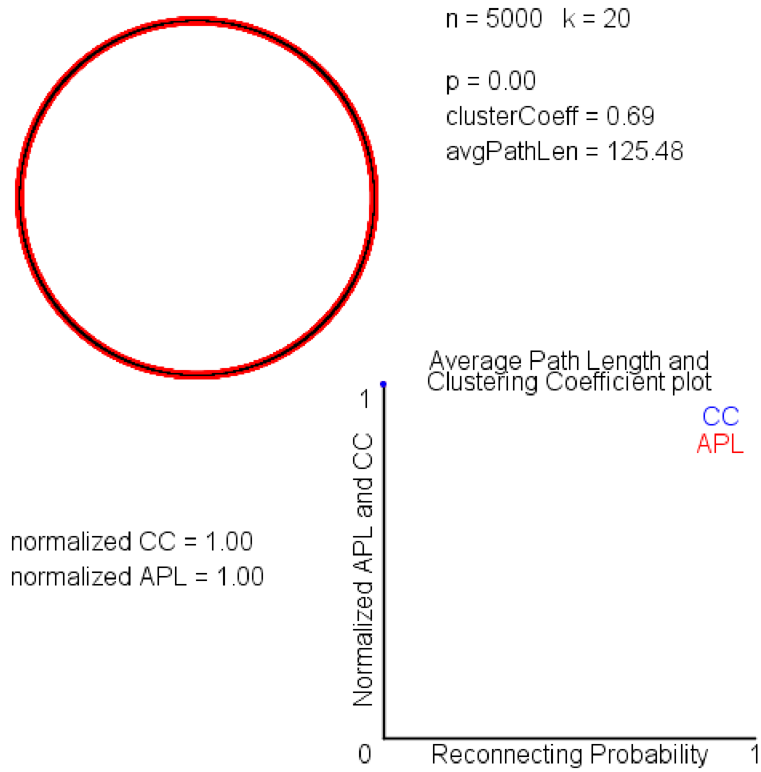

Figure 5.5: Regular graph simulation with p = 0 for n = 5000, k = 20.

Figure 5.5 ~ Figure 5.8 show the results of simulation starting with a ring of 5000 vertices, each connected to its 20 nearest neighbors by undirected edges. Figure 5.5 is a regular graph, it is the result when reconnecting possibility is 0. We see that the clustering coefficient for a regular graph with n = 5000, k = 20, p = 0 is 0.69, and the average path length is 125.48, which means the shortest distance between 2 randomly chosen nodes in the graph is 125.48. This graph, similar to Figure 5.1, has a high clustering coefficient, but the average path length is large. Figure 5.5 exhibits typical features of a regular graph: large clustering coefficient, yet far apart of nodes.

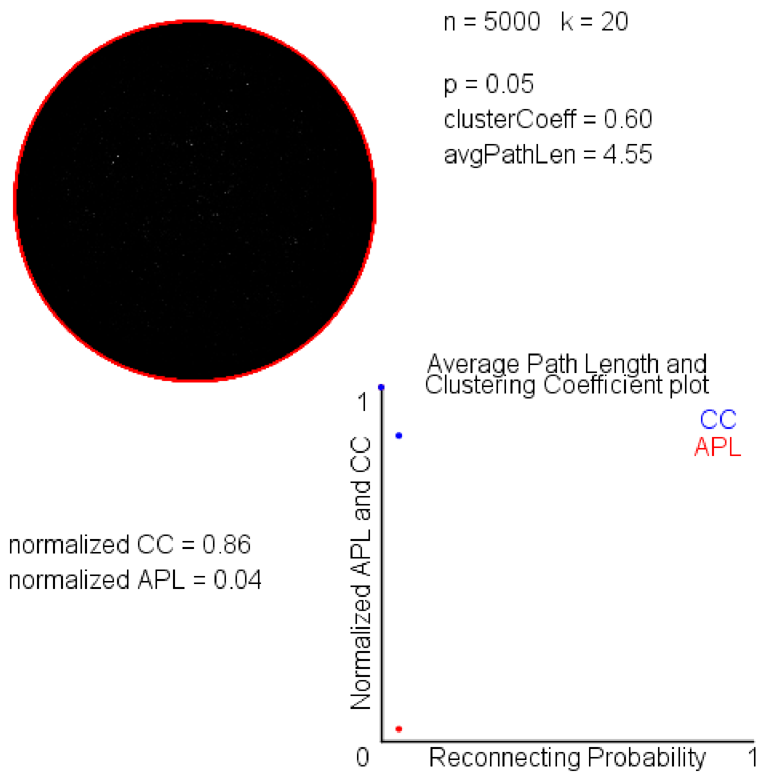

Figure 5.6: Random reconnecting simulation with p = 0.05 for n = 5000, k = 20.

Figure 5.6 is the result of simulation for random reconnecting with Reconnecting Probability of 0.05, from a regular graph with n = 5000, k = 20. This graph has a clustering coefficient of 0.6, and an average path length of 4.55. The normalized clustering coefficient is 0.86 and normalized average path length is 0.04, which means that comparing with the regular graph in Figure 5.5, clustering coefficient decreases to 86% of original value, the average path length decreases to 4% of original value! The most interesting part is the notably large percentage of decrease in the average path length. The average path length has decreased 96%! These results are constant with the observation in Figure 5.2, yet the decrease in average path length is far more significantly. These results further confirm that a little disturber (reconnecting with probability of 0.05) to a regular graph will results in a huge decrease in average path length, but has little influence on clustering coefficient. This graph exhibit typical features of a “small-world” phenomenon, having high clustering coefficient and short average path length.

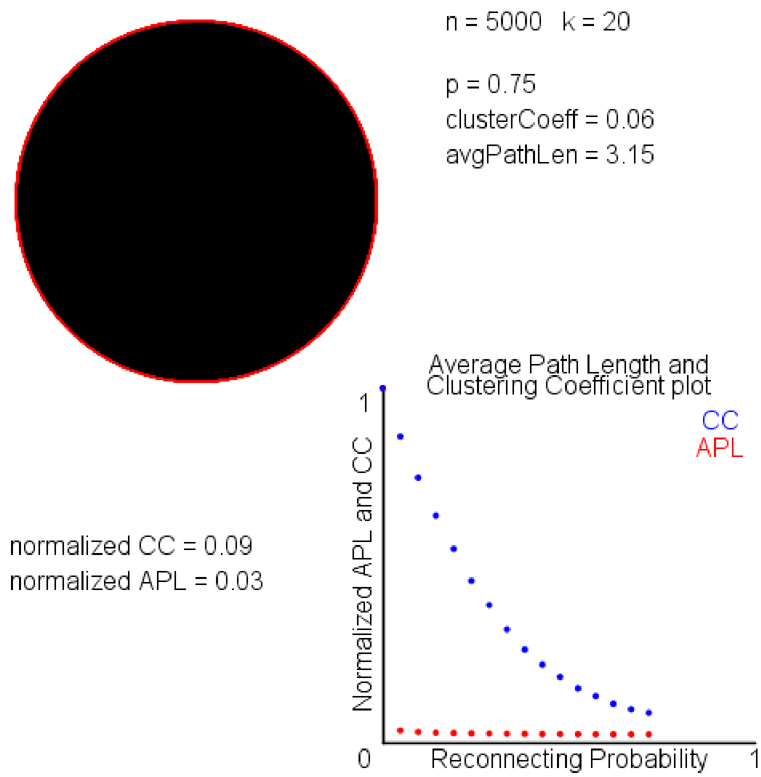

Figure 5.7: Random reconnecting simulation with p = 0.75 for n = 5000, k = 20.

Figure 5.7 is the result of simulation for random reconnecting with Reconnecting Probability of 0.75 from a regular graph with n = 5000, k = 20. This graph has a clustering coefficient of 0.06, and an average path length of 3.15. From the Average Path Length and Clustering Coefficient plot, we see that as randomness of the graph increases, clustering coefficient keeps on decreasing, but average path length remains stable after its huge drop at the beginning.

Figure 5.8 is the result of simulation for random reconnecting with Reconnecting Probability of 1.00, from a regular graph with n = 5000, k = 20. With a Reconnecting Probability of 1.00, the regular graph in Figure 5.5 has been transformed to a random graph. Each node randomly connects to other nodes. This graph has a clustering coefficient of 0.05, and an average path length of 3.13. This graph exhibits typical features of a random graph: having short average path length and small clustering coefficient.

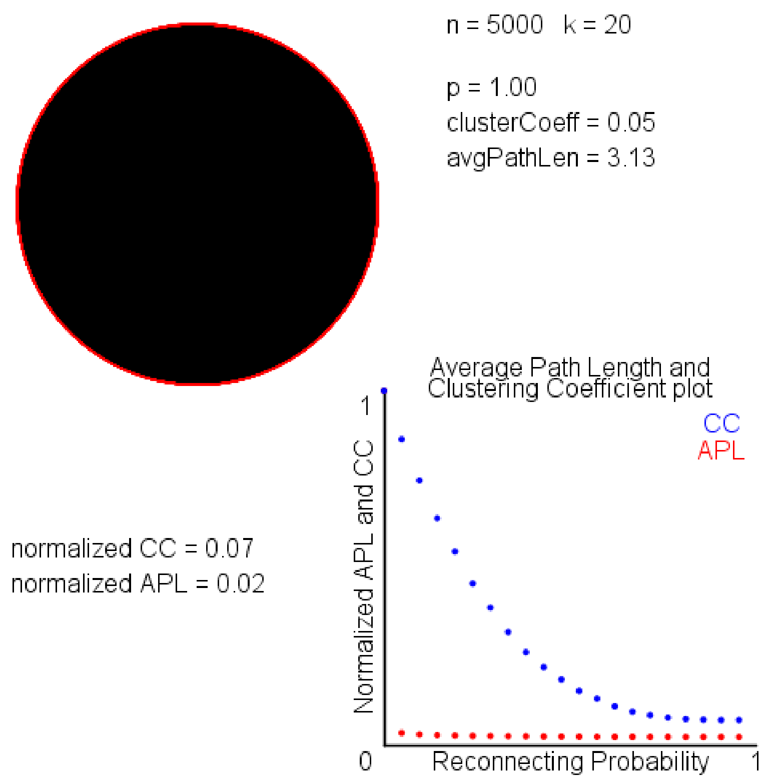

Figure 5.8: Random reconnecting simulation with p = 1.00 for n = 5000, k = 20.

6. Conclusions

We investigated the Watts-Strogatz random graph model through simulation. The Watts-Strogatz model starts with a regular graph of ring lattice, every node connects to its k nearest neighbors. The small-world graph and random graph are created from the regular graph: with probability p, rewiring each edge. By simulating graphs with node size of 100, 5000 and neighbor size of 4, 20, we observed the behaviors of regular, small-world and random graphs in terms of clustering coefficient and average path length. We observed that a regular graph have high clustering coefficient, but also high average path length. However, reconnecting a regular graph with a very small probability (< 0.1) will results in a huge decrease in average path length, but has little influence on clustering coefficient. Clustering coefficient stays high after the little disturber, but average path length fall dramatically. The graph produced by reconnecting a regular graph with small probability exhibits typical features of a "small-world" phenomenon, having high clustering coefficient and short average path length.

References

Small_World_Networks. 2015. "https://www.youtube.com/watch?v=SEOgu9GkvBw."

Watts, Duncan J, and Steven H Strogatz. 1998. "Collective dynamics of ‘small-world’networks." nature no. 393 (6684):440-442.

Wikipedia. 2015a. "http://en.wikipedia.org/wiki/Clustering_coefficient."

Wikipedia. 2015b. "http://en.wikipedia.org/wiki/Watts_and_Strogatz_model."