STATISTICAL PHYSICS OF SOCIAL NETWORK

NANYANG
TECHNOLOGICAL
UNIVERSITY

SUBMITTED
BY

LIU MEI TING, VICKY

DIVISION OF PHYSICS & APPLIED PHYSICS
SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

A final year project report
presented to
Nanyang Technological University
in partial fulfilment of the
requirements for the
Bachelor of Science (Hons) in Physics
Nanyang Technological University

April 2009

Table of Contents

Acknowledgement

I would like to express my heartfelt thanks to Professor Chew Lock Yue for his patience and guidance through this project. His knowledge and insights in Physics have been a great inspiration and support to me in the completion of this project.

I would like to thank my fellow course mates and my friends in NTU for their moral support through this year.

List of Figures

List of Tables

List of Symbols

WWW	**World Wide Web**
RGT	**Random Graph Theory**
ER Model	**Erdös-Renyi Model**
SWN	**Small-World Networks**
WS Model	**Watts-Strogatz Model**
SFN	**Scale free Networks**
APL	**Average Path Length**
APL Algorithm	**Average Path Length Algorithm**
CC	**Clustering Coefficient**
DGD	**Degree Distribution**
BA Model	**Barabási-Albert Model**

Chapter 1 Introduction

A complex network is a network graph with topological mapping features. There is a wide range of systems in nature and society that can be described by complex networks. For example, The World Wide Web is a huge virtual network of Web pages connected by hyperlinks and the Internet is a complex network of routers and computers linked by several wireless and physical cable links. In this project, virtual social networks on the Internet such as the Friendster Social network on the WWW will be studied. By constructing these networks through data gathered from the internet, the statistical physics of the resulting network topology will be explored. In particular, different network types and network properties such as Average Path Length, Clustering Coefficients and Degree Distributions will be studied. Finally the objectives and scope of the project will be presented.

1.1 The Study of Network Topology

A network is constructed by nodes and edges. A node is referred as a connection point in a network topology at which the lines intersect or branch. The connecting lines are thus referred as edges. As early as in the 18th century, the subject of complex network theory has been investigated in several areas such as sociological aspect, mathematical aspect or even in the recent years, physicists' perspective. (10)

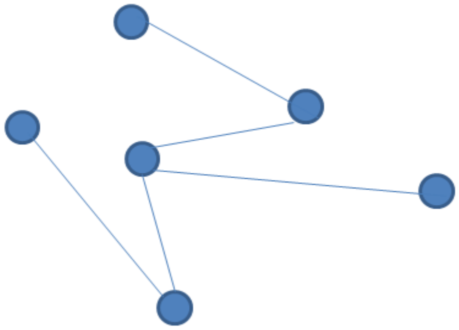


Figure 1.1 – A network with nodes and edges. Nodes are represented by blue circles and edges are represented by blue lines

In this project, investigations and discussions on the various network theories as well as the network properties will be done from the perspective of a physicist. From past literature surveys, physicists have actually inspired several works of algorithms, theories and techniques to study these network systems and its properties. In Chapter 2, the properties of Random Graph Theory, Scale-free Network Theory and Small-World Network Theory will be discussed.

1.2 Objective

As the studies for complex networks broadens, many scientists were able to investigate the various network properties, out of which the much-studied networks were mainly the Movie actor collaboration networks, World Wide Web connections and the Biological Cells networks, where properties namely Average Path Length, Clustering Coefficients and Degree Distributions were studied (10). The objective of this research is to study the network properties of a virtual social network on the Internet, and to check whether the degree distributions of these categories of social networks on the Internet exhibit possible features of a universal power law tail.

1.3 Scope of the project

In this project, a virtual social network of 204 nodes from the Friendster network on the Internet will be studied to determine its Average Path Length, Clustering Coefficients and Degree Distributions. As such, an algorithm program will be created to link and calculate the three network properties effectively using the MATLAB program. Further analysis will also be conducted for the Degree Distributions to investigate the universal power law tail of a social network.

Chapter 2 Complex Network Theories

2.1 Random Graph Theory

Traditionally, the studies of complex networks were focused in the study of random graph theory as in the 1950s large scale networks were described as random graphs. Random graphs were first studied by Hungarian mathematicians Paul Erdös and Alfred Rényi to which the Erdös-Rényi Model was proposed. In the ER Model, a random graph is defined as N nodes connected by *n* edges, which are chosen randomly from the N(N-1)/2 possible edges (Erdös and Rényi, 1959). Alternatively, this Model can be also be deemed as a binomial model, making use of connection probability *p* in the understanding of network theory, which can be further understood from the journal “Statistical mechanics of complex networks” by Réka Albert and Albert-László Barabási (10). Results in this project will thus reveal the probability of appearances of subgraphs, cycles and trees in the targeted social network, which will then be further analyzed in Chapter 4. As such, as the size of networks increases, that is the number of nodes increases, the number of neighboring nodes also increases with the network size.

2.1.1 Average Path Length of Random Graph Theory

The diameter of a graph is the maximal distance between any pair of its nodes. Random Graphs tend to have small diameters, provided the probability *p* is not too small. This is due to the fact that a random graph is likely to a spreading graph (10). One method adopted to study the spread of a random graph is to calculate the average distance between any pairs of nodes, which is the Average Path Length. In this context, distance refers to the shortest path between any 2 nodes. In the ER Model, the average path length is defined to scale with the number of nodes in the following: $l_{rand} \sim \frac{\ln(N)}{\ln(\langle k \rangle)}$.(Wandora 2009) This equation was further analyzed in the journal paper "Statistical Mechanics of complex networks" by Réka Albert and Albert-László Barabási, to which the graph in comparison between average path length of real networks and the prediction was plotted to reflect evidence that average path length of real networks is close to average path length of random graph of the same size (10).

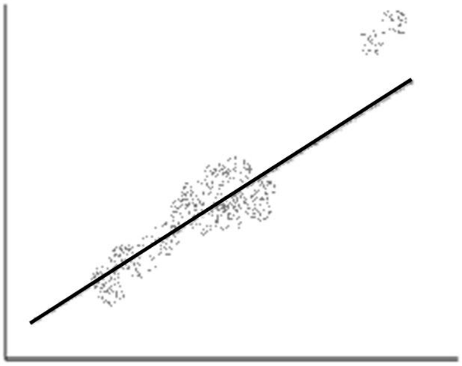


Figure 2.1 – Schematic Plot of FIG. 8. (10)
Comparison between the average path lengths of real networks

With reference to FIG. 8 of the paper, the APL of a social network will therefore be analyzed and compared to the APL of Random Graph Theory in Chapter 4 of this thesis.

2.1.2 Clustering Coefficient of Random Graph Theory

Theoretically, complex networks exhibit a large degree of clustering (2). In relation to RGT and Probability theory, consider a node in a RGT together with its connecting neighbors, the probability that two of these neighbors are connected is equal to the probability that two randomly selected nodes are connected. Consequently the clustering coefficient of a random graph (Wandora, 2009) is

$$C_{rand} = p = \frac{\langle k \rangle}{N}$$

Theoretically, taking the plot of the ratio $C_{rand}/\langle k \rangle$ as a function of N for random graphs of different sizes, on a log-log scale a straight line of slope -1 will be obtained.

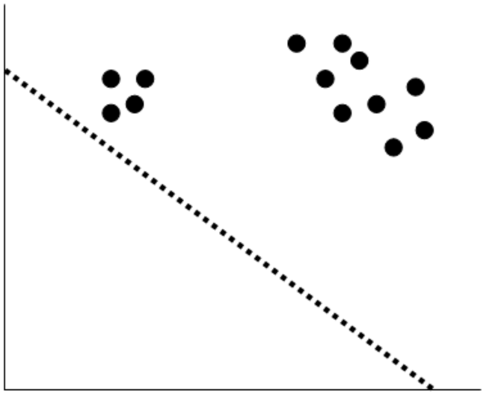


Figure 2.2 – Schematic Plot of FIG. 9. (10)
Comparison between CC of Real Networks and random graphs.

To verify the authenticity of this theory, FIG.9 of "Statistical mechanics of complex networks" shows the plot of the ratio of the clustering coefficient of real networks and their average degree, $\langle k \rangle/N$ as a function of their nodes size. (10) Here, degree is defined to be the number of connections or edges the node has to other nodes. So, from the social network perspective, the degree is the number of friends a person has.

2.1.3 Degree Distribution of Random Graph Theory

In the ER Model, a random graph with connection probability *p* is said to have its degree k_i of a node to follow a binomial distribution such that $P(k_i = k) = C_{N-1}^{k} p^k (1-p)^{N-1-k}$. That means, the probability of a node having k edges is p^k.

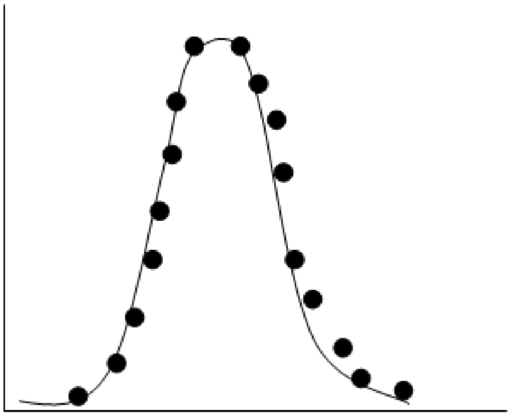


Figure 2.3 – Schematic Plot of FIG. 7. (10)
Degree Distribution of random graph

For a random graph, it has been found that DGD of a random graph follows a Poisson Distribution as shown in FIG. 7 of “Statistical mechanics of complex networks”. (10) Therefore, Chapter 4 of this thesis will study the DGD properties of the social network to see whether a Poisson distribution graph is exhibited.

2.2 Small-World Network Properties

2.2.1 The Watts-Strogatz Model

A Small-World network is a network that captures the small world phenomenon of strangers being linked together by a mutual acquaintance. In this context, the nodes represent people and the edges represent the relation that people know each other.(10) The Watt-Strogatz Model, first proposed by Strogatz and Watts, had originated in social systems in which most people are friends with their immediate neighbors, neighbors here implies immediate colleagues, classmates or even relatives etc. However, everybody tends to have one or two friends that are a long way away such as people in other countries or old friends, which are then represented by long-range edges obtained by rewiring edges in the WS Model.(10) At this point, a similarity can be observed between this WS Model and the Friendster social network that is being studied such that the “friends” in this network are connected to each other due to being immediate neighbors even in real life, and that, there are also certain “friends” in the network which are long-range edges as described in the WS Model. This model is then easier to analyze because it does not involve the formation of isolated clusters, which happens in the original WS Model as discussed by Réka Albert and Albert-László Barabási.

2.2.2 Average Path Length of Small-World Networks

Average path length is one of the most robust measures in network topology. It is defined as the average number of edges along the shortest paths for all possible pairs of network nodes. In Small-World Networks, one question often asked about the APL was whether the Small-World characteristic is dependent on the size of the network. Studies have thus showed that the APL of Small-World Networks is independent on the network size and that Small-World Networks have low APL due to the small-world phenomenon.

2.2.3 Clustering Coefficient of Small-World Networks

An important factor of Small-World Networks is that they have large clustering coefficient. Further elaborated by Réka Albert and Albert-László Barabási, in a regular lattice (p=0) the clustering coefficient does not depend on the size of the lattice but only on its topological features. While the edges of the network are randomized, the clustering coefficient remains close to C(0) up to relatively large values of p. The dependence of C(p) on p can be derived using a slightly different but equivalent definition of C, introduced by Barrat and Weigt (2000). According to this definition, C'(p) is the fraction between the mean number of edges between the neighbors of a node and the mean number of possible edges between those neighbors.(6) As such, Small-World Networks are said to exhibit large clustering coefficients. With respect to the above formulation introduced in the journal paper, comparisons between the different formulations of the clustering coefficients will be taken into consideration in the analysis in Chapter 5 of this thesis.

2.2.4 Degree Distribution of Small-World Networks

In the WS Model, for *p*=0, each node has the same degree K. Thus the degree distribution is the delta function centered at K. A nonzero connection probability *p* will thus introduce more degrees into the network, thus widening the degree distribution with the larger range of degrees in the network (6). At the same time, Réka Albert and Albert-László Barabási maintains the argument that the broadening of the degree distribution maintains the average degree to equal a particular value K. (10) Therefore, It was further explained that for K>2 the network is usually connected which differs from random graph that consists of isolated clusters when there is wide range of connection probability.(10)

2.3 Scale Free Networks

A scale free network is a network whose degree distribution follows a power law. (11) This is also the main property that determines whether a certain network can be classified as a scale-free network. Further discussions by Réka Albert and Albert-László Barabási explains that random graph theory and the WS Model cannot reproduce the power-law distribution feature, which thus introduced the Barabási-Albert Model, who argued that the scale-free nature of real networks have properties shared by several real networks (11).

Chapter 3 Experimental Approach

3.1 Average Path Length

Before determining the Average Path Length of a social network, the method to calculate will be studied. Consider a network of 6 nodes that is fully connected as shown in Figure 3.1.

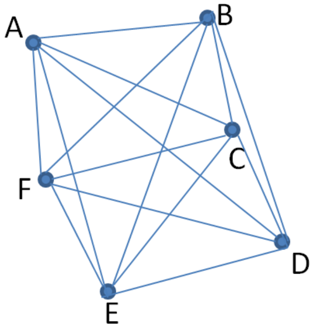


Figure 3.1 – A fully connected network with N=6

Starting from the first node denoted by A, it will be fully connected to the rest of the nodes by 5 edges as shown. For N=6, the edges connected to node A will be (N-1) edges. A pattern can be traced in the completion of the edges to each node whereby the number of edges needed to connect to the subsequent nodes will decrease by 1 each time. As such, the remaining number of edges needed to fill the second node, B to the rest of the nodes will be 4 edges, which is (N-2) edges and third node, C will be 3, which is (N-3) edges, until the last node will be left with only the remaining last edge, which is (N-5) edges to produce a fully connected graph as shown in Figure 3.2. In the mathematical context, this pattern can be inferred as the Arithmetic Series for the formulation for the maximum number of edges in a particular network. To determine the maximum number of edges for this network for each node, we take the sum of the numbers of edges connected to each node, not repeating the nodes that had been added to the previous nodes, which gives us,

Maximum Edges for 6 nodes = (N-1) + (N-2) + (N-3) + (N-4) + (N-5) +1 = 15

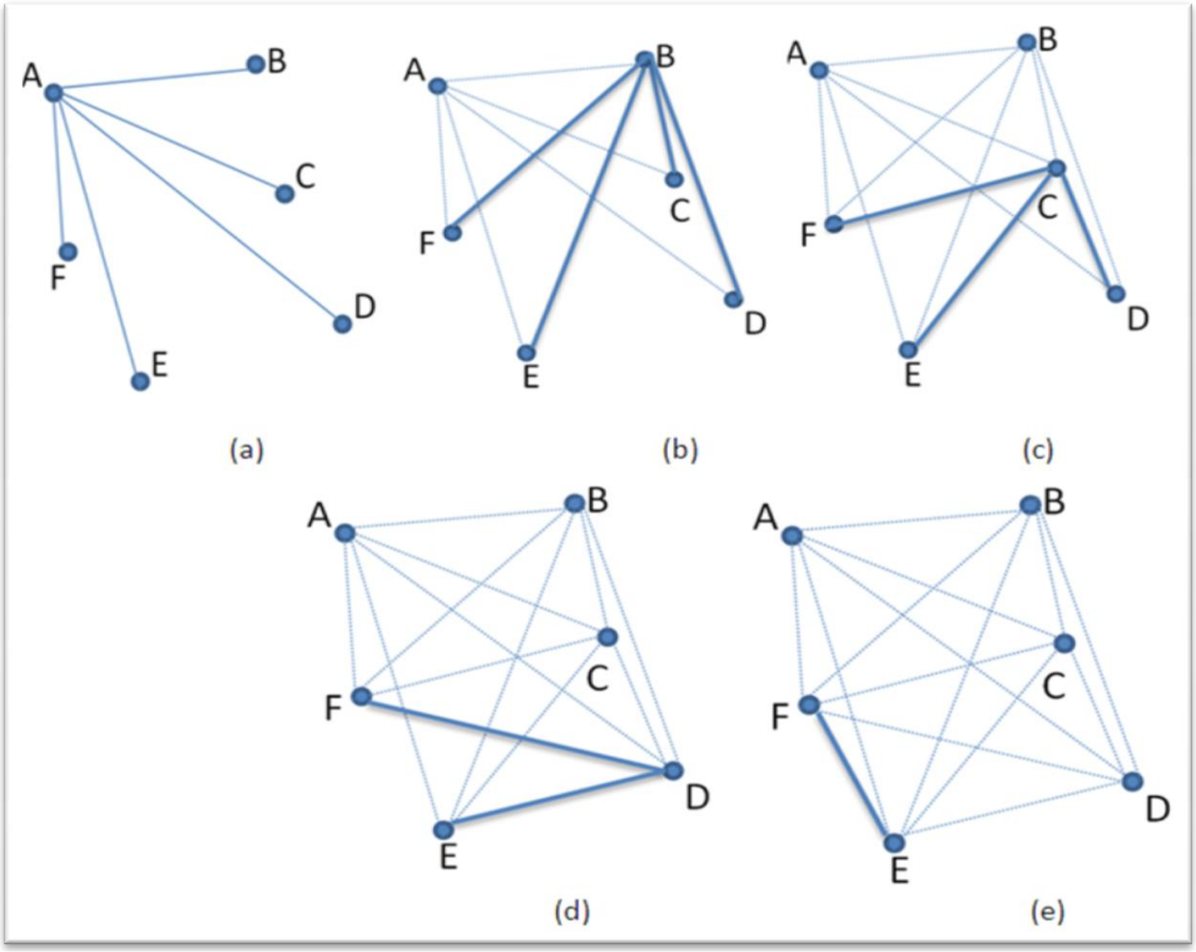


Figure 3.2 – illustration of the connection of edges to subsequent nodes in network. (a) Node A has 5 connecting edges. (b) Node B has the remaining 4 edges to connect. (c) Node C has the remaining 3 nodes to connect. (d) Node D has the remaining 2 nodes to connect. (e) Node E has the last node to connect.

Based on this pattern in Figure 3.2, this maximum edges formulation is similar to the one derived in the Erdös-Rényi Model where maximum edges = [N (N-1)]/2.

3.1.1 Average Path Length Derivation

In Chapter 2, various formulations in obtaining the Average Path Length for the different network theories were introduced and discussed. In this project, Average Path Length of a network is defined as the average path taken from the sum of all the shortest paths between all pairs of nodes in a network. As such, the shortest path is the shortest number of steps or edges for a node to reach another node. Consider a network of 5 nodes as shown in Figure 3.3, known as Network Star, randomly connected with 6 edges. The maximum number of edges possible for Network Star is therefore 10 edges.

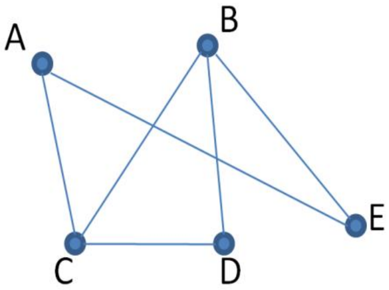


Figure 3.3 Illustration of Network Star

To calculate the Average Path Length for this network, the shortest path taken for every pair of nodes is first computed and formed as in Table 3.1.2.

A → B = 2	B → C = 1	C → D = 1	D → E = 2	E → A = 1
A → C = 1	B → D = 1	C → E = 2	D → A =2	E → B = 1
A → D = 2	B → E = 1	C → A = 1	D → B =1	E → C = 2
A → E = 1	B → A = 2	C → B = 1	D → C = 1	E → D = 2

Table 3.1.2 – Table of Shortest Path Data

From Table 3.1.2, a property of network theory is reflected which differentiates directed graphs and undirected graphs. A directed network graph is a graph which the edges have direction between nodes. Compared to undirected graph shown in Figure 3.4(b), the Average Path Length for an identical network for directed graph and undirected graph will thus differ in value due to the orientation of the edges towards the nodes. As such, emphasis was placed in calculations for undirected graphs. Therefore, only the group of shortest path data in the either the upper or lower diagonals of the values in Table 3.1.2 need to be considered, that is taking the sum of the values highlighted in blue or in yellow. (Either one give the same summation)

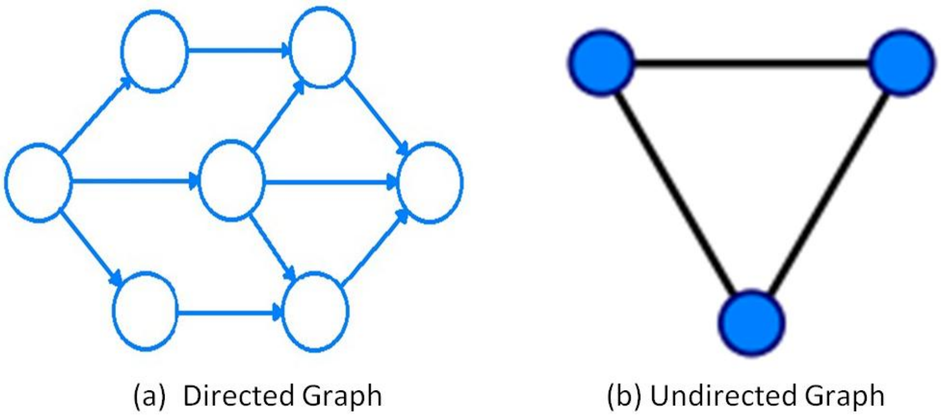


Figure 3.4- Illustration of Directed and Undirected Graphs

To determine the Average Path Length of the network in Figure 3.3, the Average Path Length is worked out as follows:

$$APL\ of\ Network\ Star = \frac{sum\ of\ shortest\ paths\ data}{maximum\ number\ of\ edges}$$

$$APL\ of\ Network\ Star = \frac{2+1+2+1+1+1+1+1+2+2}{\frac{5(5-1)}{2}} = \frac{14}{10} = 1.4$$

With this formulation of the Average Path Length, the APL Algorithm will thus be implemented according to this formulation to determine the Average Path Length of networks for this project.

3.1.2 Average Path Length Data Computation

Before we can calculate the shortest paths needed to reach each node in Table 3.1.2, the method to count the shortest path is taken into consideration for the APL Algorithm. A method was adopted such that “Network” data consist of a square array of measurements. The rows are the cases, or subjects. The columns of the array are the same set of cases or subjects linked to the other cases in the same network. In each cell of the array describes a relationship between the subjects. Applying this method in the subject of social networks, consider nodes that have direct connection with edges in the network, “1” indicates there is a direct edge connection between a particular pair of nodes and “0” indicates no direct edge connection between the pair of nodes, that is, there have to be more than 1 edge and the node have to be linked to another node to be able to reach the targeted node in the network.

Nodes	A	B	C	D	E
A	0	0	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Table 3.1 – Array of data with direct edge connections

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	2	0	1	1	1
C	1	1	0	1	2
D	2	1	1	0	2
E	1	1	2	2	0

Table 3.2 – Array of data with direct and indirect edge connections

As such, Table 3.1 displays the array of data for direct edges connection in Network Star (Figure 3.3). Notice the set of data in Table 3.1, the cells (1,1), (2,2), (3,3) and (4,4) are left as "0" because the node cannot be connected to itself. In the Algorithm proposed this diagonal of the matrices neither affect the final results nor is involved in any calculations. However, Table 3.1 is insufficient such that it is only relevant in determining whether any particular pairs of node have direct edge connection, it does not compute the number of edges to which the indirectly connected nodes are connected. By manual calculation the complete array of data which includes the shortest paths for every pair of nodes can be found in Table 3.2. Therefore, the concept of the APL Algorithm is to update Table 3.1 to the set of cell array data in Table 3.2 to include data of direct edge and indirect edge connections values.

3.1.3 Average Path Length Algorithm

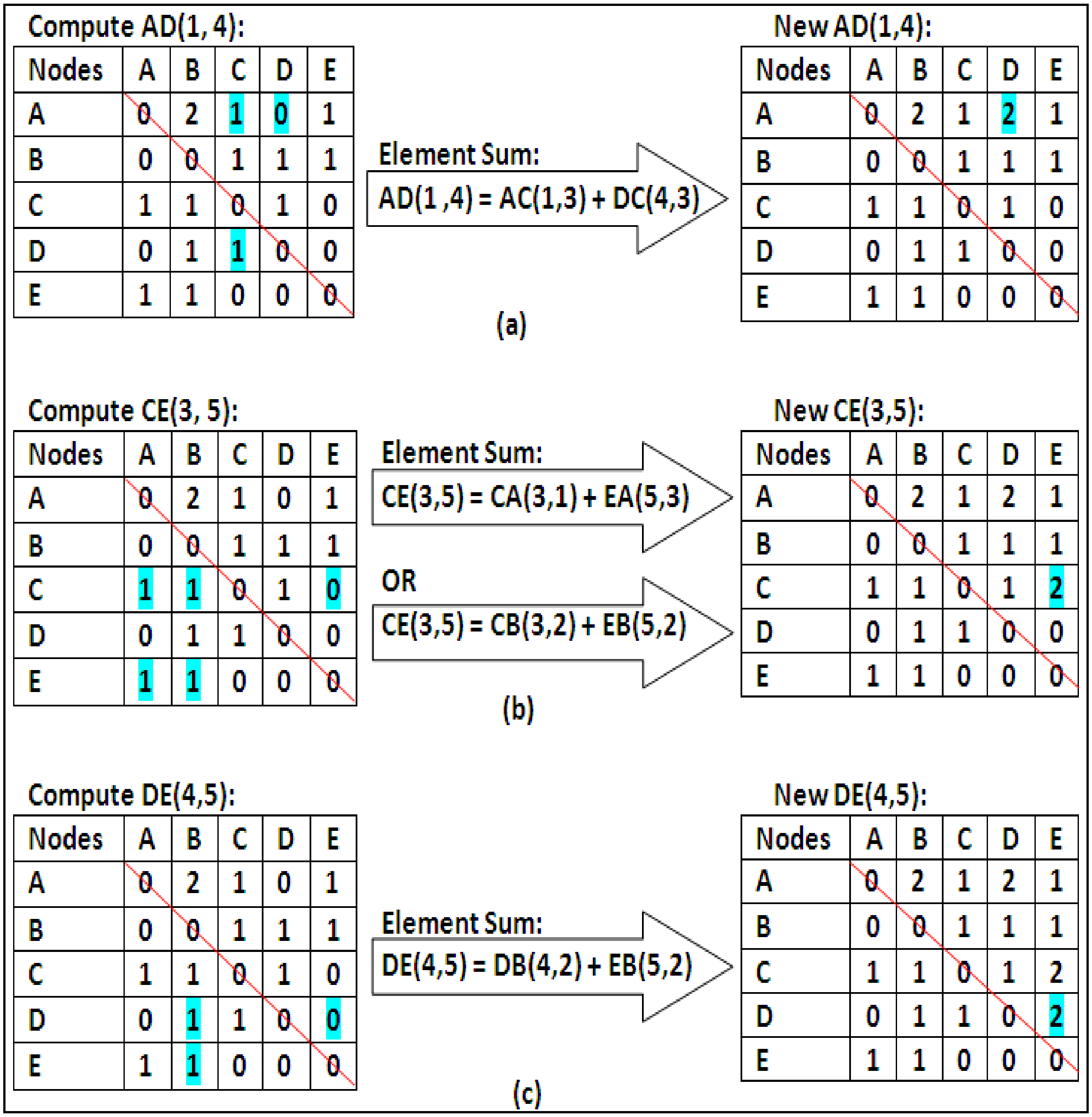
We try to find a pattern between the transitions from “0”s into values which reflect the shortest paths taken for the particular nodes connections. With reference to Network Star, the shortest path taken for node A to be connected to node B will be studied first. The value “0” in cell AB(1, 2) reveals A is not directly linked to B with an edge. From Figure 3.3, we observed that the shortest path needed is 2 edges, with two alternatives of connecting A and B, one A→C→B and the other A→ E →B. Subsequently, an interesting pattern shows that with AB(1, 2) of the cell array being the focus.

1. Search across the row of the subject (Row 1 of Network Star Table 3.5). Notice that cells AC(1, 3) and AE(1,5) are both labeled as “1” and are also coincidentally the associated links between nodes A and B in Figure 3.3.
2. Emphasis is placed on the columns involved which is columns 3 and column 5 and another similarity is observed, where the indication of “1” is across Row 2(cells BC(2,3) and BE(2,5)), which reflects the direct connection to Node B.
3. As such, in mathematical terms a direct summation of the “1”s in the targeted Rows and Columns for the targeted nodes is taken to calculate the shortest paths for the nodes with no direct edges linked.
4. Technically, computation of the element sum of the cell values in the pattern produces 2 statements: AB(1,2) = AC(1,3) + BC(2, 3) and/or AB(1,2) = AE(1,5) + BE(2, 5).

In view of the above computation, the same method to compute the remaining nodes is applied. In addition, the display of the data in the cell array format also has an interesting feature such that the upper triangular matrix data is a direct transpose of the lower triangular matrix. By saying it as a transpose means the value reflected in the cell (x, y) is of the same value as cell (y, x). Working with transposes, we shall focus on the upper triangular matrix to conduct future calculations. Figure 3.5 thus show various observations from the computation of different nodes that will be taken into account in the forming of the APL Algorithm.

1. Computation of cell AD(1,4) requires the summation of values from cells AC(1,3) and DC(4,3), and both values falls in the upper and lower triangular array respectively.
2. Computation of cell CE(3,5) requires the summation of values either cells CA(3,1) and EA(5,3) or cells CB(3,2) and EB(5,2) and all values fall in the lower triangular array.
3. Computation of cell DE(4,5) requires summation of cells DB(4,2) and EB(5,2) and both values fall in lower triangular array.
4. Comparing the method of acquisition of values from the array, it would be more efficient to obtain values solely from either the lower or upper triangular array in order to make the APL Algorithm more organized and systematic when being applied through large networks.
5. As such, the APL Algorithm is organized in a way that the values will be only taken and computed from the upper triangular array before being updated as a transpose to the lower triangular array.

Therefore, the revised version of the APL Algorithm will be programmed according to the method as shown in Figure 3.6 to reflect a more organized way of retrieving and computing of values.



Compute AD(1, 4):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

AD(1 ,4) = AC(1,3) + DC(4,3)

New AD(1,4):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

(a)

Compute CE(3, 5):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

CE(3,5) = CA(3,1) + EA(5,3)

OR

CE(3,5) = CB(3,2) + EB(5,2)

New CE(3,5):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	2
D	0	1	1	0	0
E	1	1	0	0	0

(b)

Compute DE(4,5):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

DE(4,5) = DB(4,2) + EB(5,2)

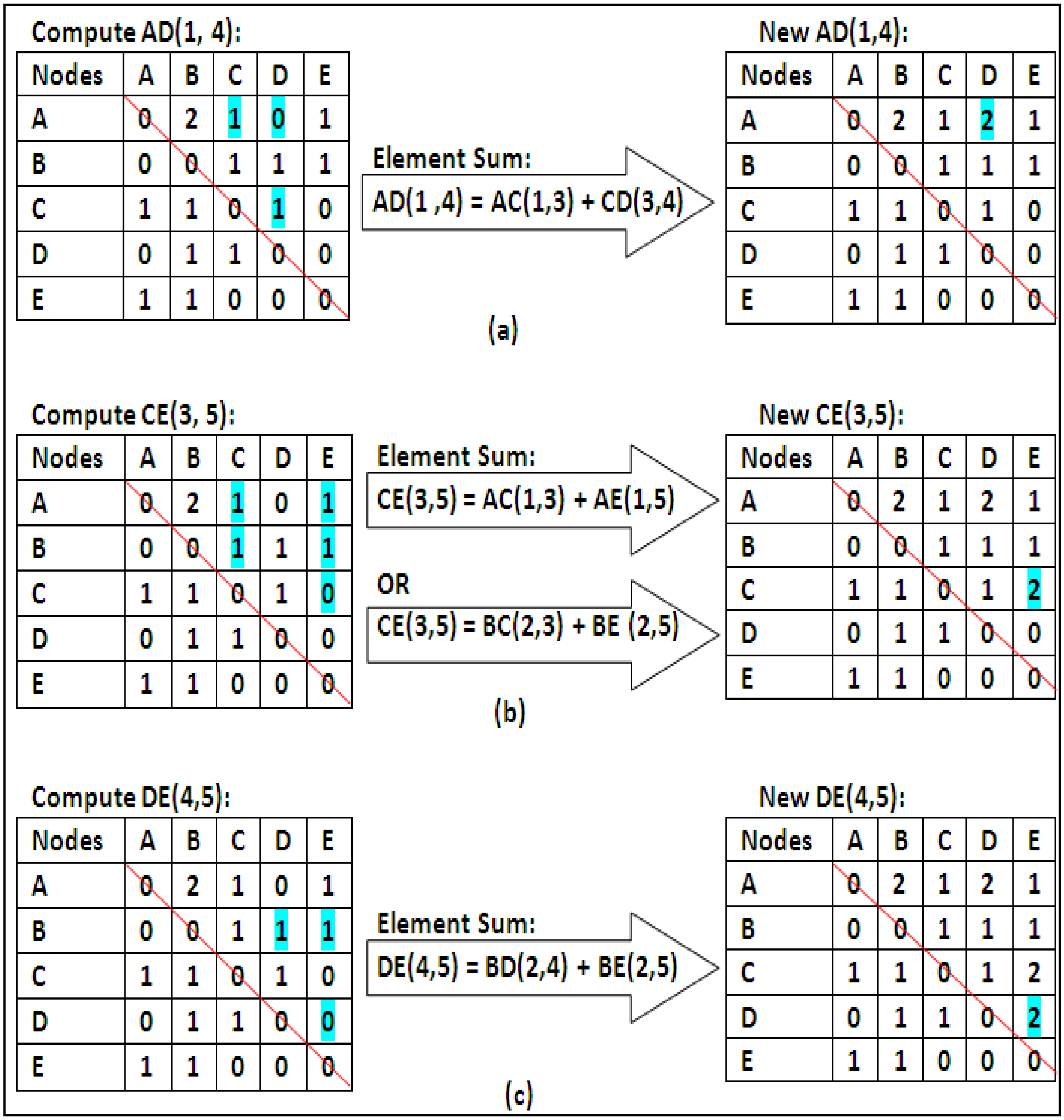
New DE(4,5):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	2
D	0	1	1	0	2
E	1	1	0	0	0

(c)

Figure 3.5 – Illustration of mathematical computation of shortest paths for nodes without direct edges. (a) Compute cell AD(1,4): values taken falls in upper and lower triangular array.

(b) Compute cell CE(3,5): values taken falls in lower triangular array and have 2 alternatives. (c) Compute cell DE(4,5): values taken falls in lower triangular array



Compute AD(1, 4):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

AD(1 ,4) = AC(1,3) + CD(3,4)

New AD(1,4):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

(a)

Compute CE(3, 5):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

CE(3,5) = AC(1,3) + AE(1,5)

OR

CE(3,5) = BC(2,3) + BE (2,5)

New CE(3,5):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	2
D	0	1	1	0	0
E	1	1	0	0	0

(b)

Compute DE(4,5):

Nodes	A	B	C	D	E
A	0	2	1	0	1
B	0	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	0
E	1	1	0	0	0

Element Sum:

DE(4,5) = BD(2,4) + BE(2,5)

New DE(4,5):

Nodes	A	B	C	D	E
A	0	2	1	2	1
B	0	0	1	1	1
C	1	1	0	1	2
D	0	1	1	0	2
E	1	1	0	0	0

(c)

Figure 3.6 – Illustration of mathematical computation of shortest paths for nodes without direct edges in the upper triangular array.

In addition to the improved APL Algorithm version in Figure 3.6, notice the element sum of the cells can be followed in either of the two patterns as follows:

Cell (x, y) = cell (x, z) + cell (z, y)

Cell (x, y) = cell (z, x) + cell (z, y)

Note that a condition of “z” value stands where z<x and z<y if and only if x≠1 or y≠1. This is because emphasis is placed only on the upper triangular array. If x or y is of the value 1 then value of z will definitely be either z>x or z>y. Lastly, to compute APL of a network is to take the element summation of the whole upper triangular array over the maximum number of edges possible for this network, based on the size of the network., which is therefore implemented in the last part of the APL Algorithm.

3.2 Clustering Coefficient

3.2.1 Clustering Coefficient Derivation

The definition of the Clustering Coefficient is based on the ratio of the number of connections in the neighborhood of a node and the total number of connections if the neighborhood is fully connected (Wandora 2009). Consider Network Star in Figure 3.3, to calculate the CC of node A, as shown in Figure 3.7, the number of connections in the neighborhood of the node implies the number of edges linked between nodes that node A have direct edges linked to, which is node C and node E.

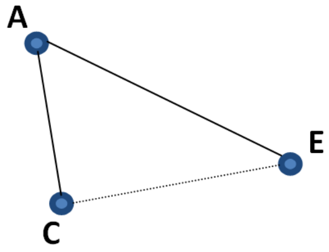


Figure 3.7 – Illustration of Node A and its neighborhood in Network Star

In this case, there is no connection between node C and E in Network Star which thus makes the numerator zero. The total number of connections if the neighborhood is fully connected implies the maximum number of edges possible if the nodes that are linked to the target node are fully connected. In this context, there are two nodes linked to node A, making the neighborhood of node A to be 2. Therefore the maximum number of edges possible for 2 nodes is 1. But the CC of A is zero as the numerator is zero.

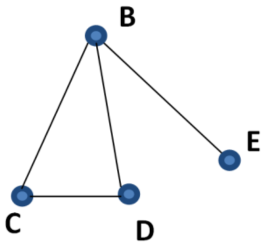


Figure 3.8 – Illustration of Clustering Coefficient computation of Node B

To calculate the CC of B, consider the neighborhood of B, the number of nodes in its neighborhood is 3 (nodes C, D and E). The number of connections between these 3 nodes is 1 (direct edge connection between nodes C and D), which makes the numerator 1(Figure 3.8). Since the nodes in the neighborhood of node B is 3, the maximum number of edges possible for 3 nodes is 3. The computation of CC of node B is thus $\frac{1}{3}$. The range of clustering coefficient therefore can only be between 0 to 1, CC value of 1 reflects the node and its neighborhood are fully connected. For example, the CC of node D of Network Star in Figure 3.8 is 1 because the number of edges connected between the neighborhood of node D (nodes C and B) is 1 and the maximum number of edges possible for 2 nodes is 1, therefore makes the CC of node D = 1. As such, the Clustering Coefficient Algorithm to be programmed will be based on the computation and derivation done earlier.

3.2.2 Clustering Coefficient Data Computation

As the nature of clustering coefficient involves direct edge connections between nodes, the cell array used in Section 3.1.3 Table 3.1 can be used again in this case to further compute and study the CC of individual nodes as well as of the network. Figure 3.9 introduces a new network of 6 nodes, named Network Hex. Figure 3.10 and Table 3.3 illustrates the manual computation of the CC of the Network Hex. Following that, the mathematical approach will be showed to compute the CC of the network as in the MATlab program.

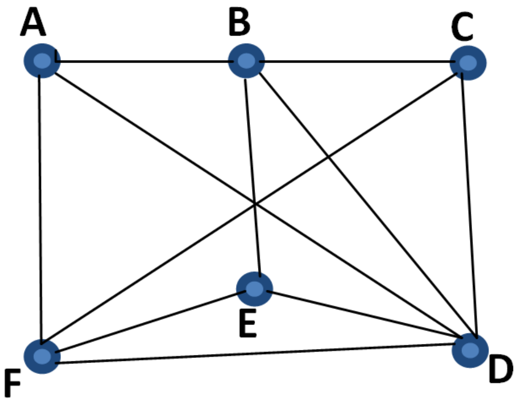
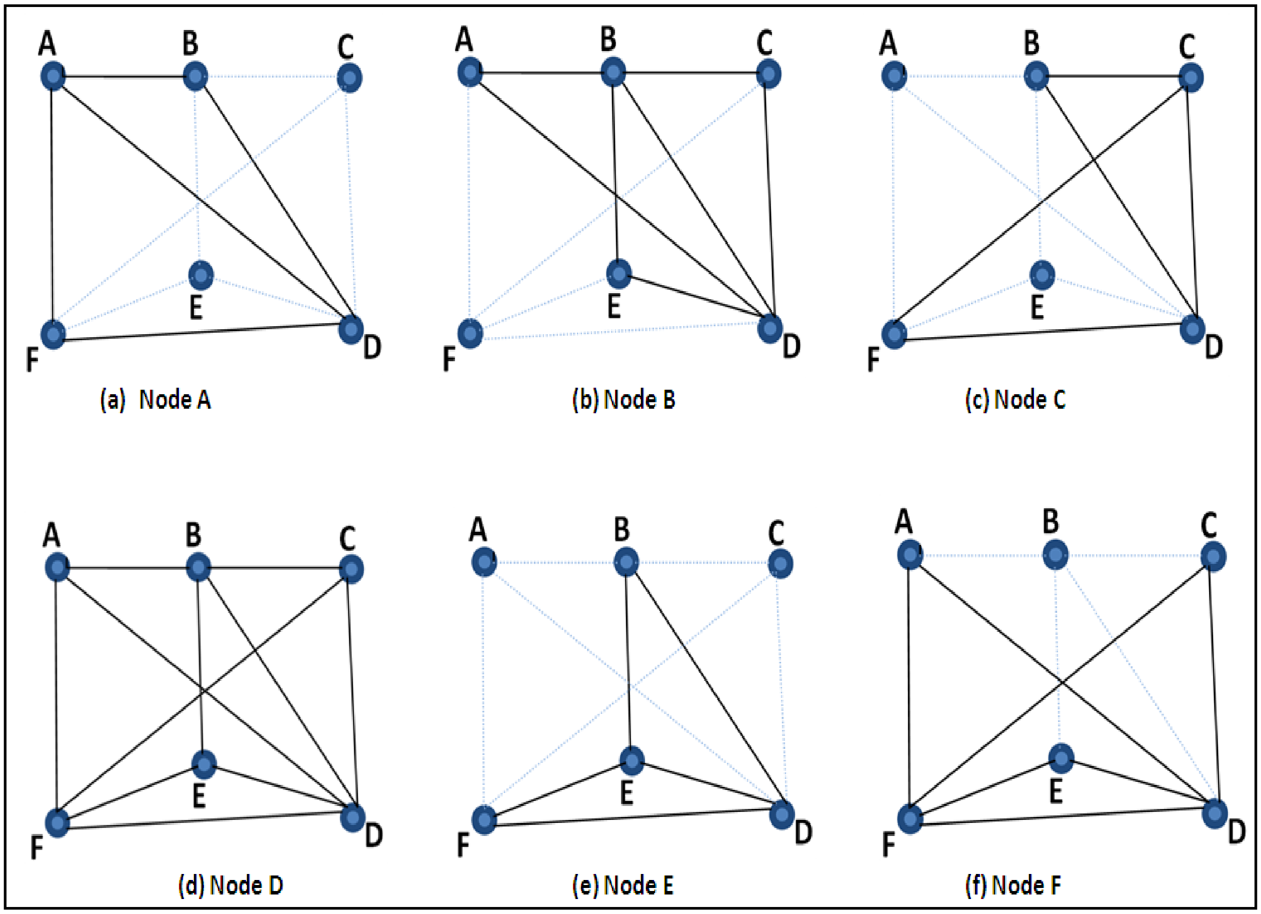


Figure 3.9 – Illustration of Network Hex, with 6 nodes



(a) Node A (b) Node B (c) Node C

(d) Node D (e) Node E (f) Node F

Figure 3.10 – Illustration of CC of individual nodes in Network Hex. Black lines indicate edges that are directly linked to the nodes.

Node	Neighbor Nodes	Direct Edge connection (Numerator)	Max Possible Edges (Denominator)	Clustering Coefficient of Node
A	3	2	3	$\frac{2}{3}$
B	4	2	6	$\frac{4}{6}$
C	3	2	3	$\frac{2}{3}$
D	5	6	10	$\frac{6}{10}$
E	3	2	3	$\frac{2}{3}$
F	4	3	6	$\frac{3}{6}$

Table 3.3 – Manual calculation of clustering coefficient of Network Hex

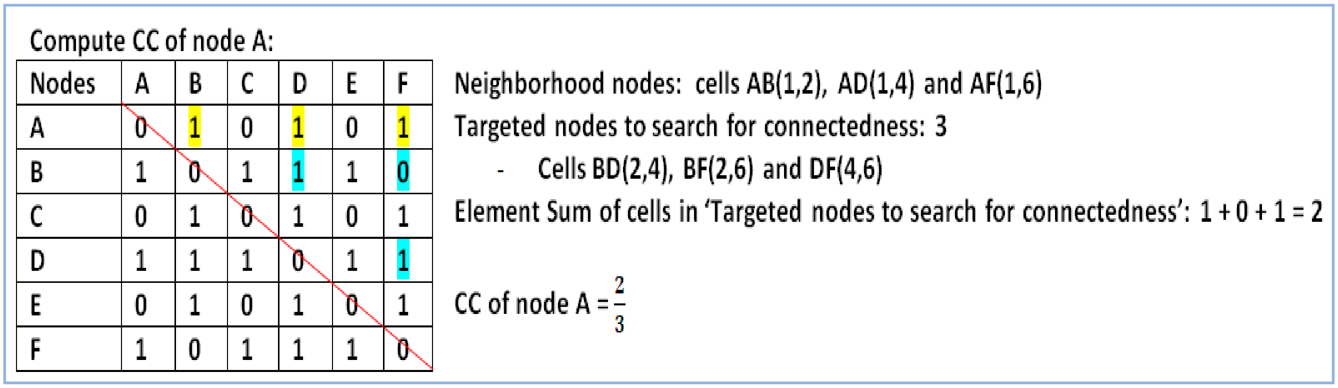
With respect to Table 3.3, the CCs of all nodes are tabulated. The Clustering Coefficient of Network Hex is therefore the average of the sum of the CCs in the network, which gives:

$$CC\ of\ network = \frac{\frac{2}{3}+\frac{4}{6}+\frac{2}{3}+\frac{6}{10}+\frac{2}{3}+\frac{3}{6}}{6} = 0.62777$$

In view of the above computation, this will be the approach attempted in finding all the Clustering Coefficients in this project.

3.2.3 Clustering Coefficient Algorithm

As mentioned earlier, using the same cell array the Clustering Coefficient can be accurately computed and presented. From Table 3.1, as defined earlier, the value "1" indicated a direct edge connection between pairs of nodes. Therefore, based on the method of computation in the previous section, we can refer to Table 3.4 to Table 3.9 for the CC Algorithm representation.



Compute CC of node A:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells AB(1,2), AD(1,4) and AF(1,6)

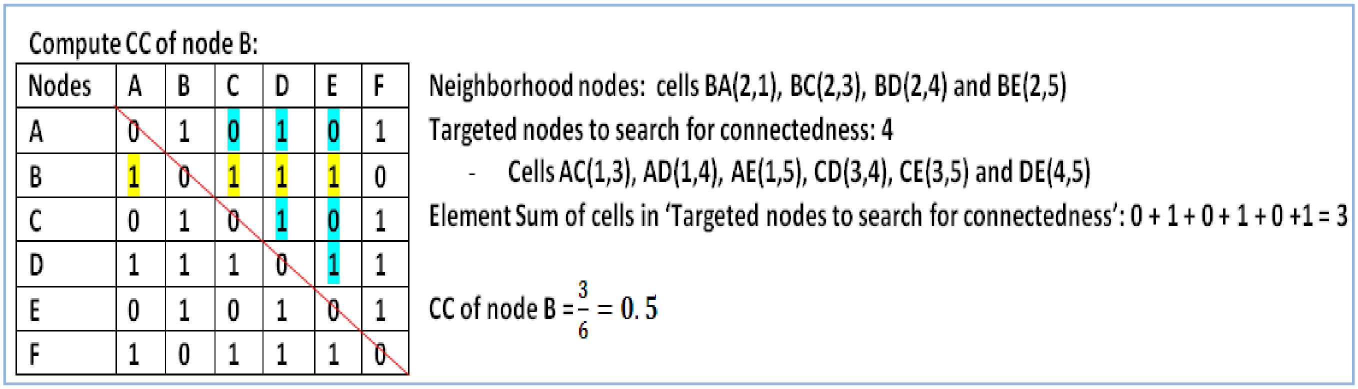
Targeted nodes to search for connectedness: 3

- Cells BD(2,4), BF(2,6) and DF(4,6)

Element Sum of cells in 'Targeted nodes to search for connectedness': 1 + 0 + 1 = 2

CC of node A = $\frac{2}{3}$

Table 3.4 – CC Algorithm representation of node A



Compute CC of node B:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells BA(2,1), BC(2,3), BD(2,4) and BE(2,5)

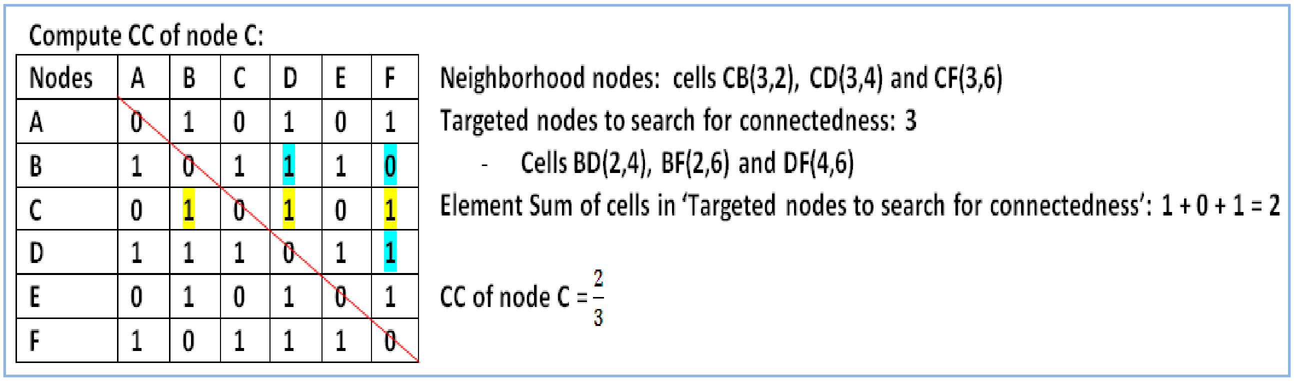
Targeted nodes to search for connectedness: 4

- Cells AC(1,3), AD(1,4), AE(1,5), CD(3,4), CE(3,5) and DE(4,5)

Element Sum of cells in 'Targeted nodes to search for connectedness': 0 + 1 + 0 + 1 + 0 +1 = 3

CC of node B = $\frac{3}{6} = 0.5$

Table 3.5 – CC Algorithm representation of node B



Compute CC of node C:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells CB(3,2), CD(3,4) and CF(3,6)

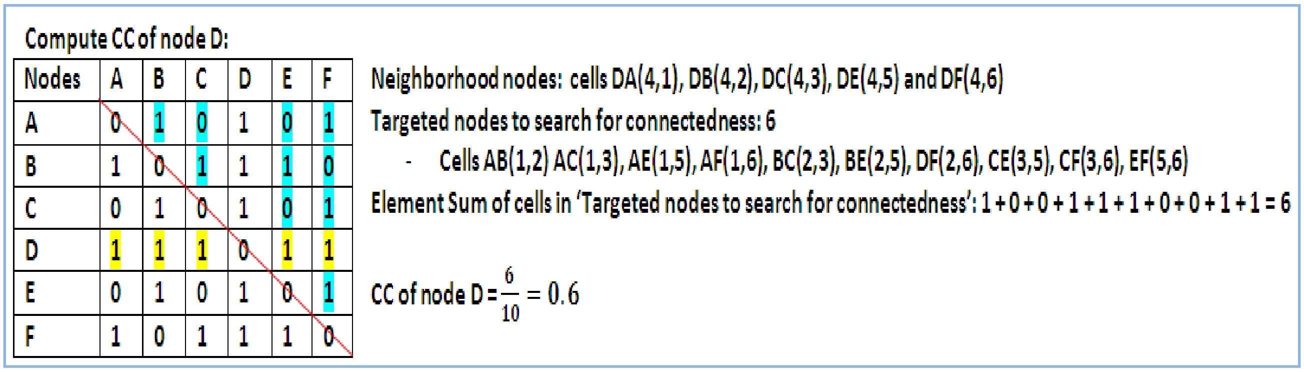
Targeted nodes to search for connectedness: 3

- Cells BD(2,4), BF(2,6) and DF(4,6)

Element Sum of cells in 'Targeted nodes to search for connectedness': 1 + 0 + 1 = 2

CC of node C = $\frac{2}{3}$

Table 3.6 – CC Algorithm representation of node C



Compute CC of node D:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells DA(4,1), DB(4,2), DC(4,3), DE(4,5) and DF(4,6)

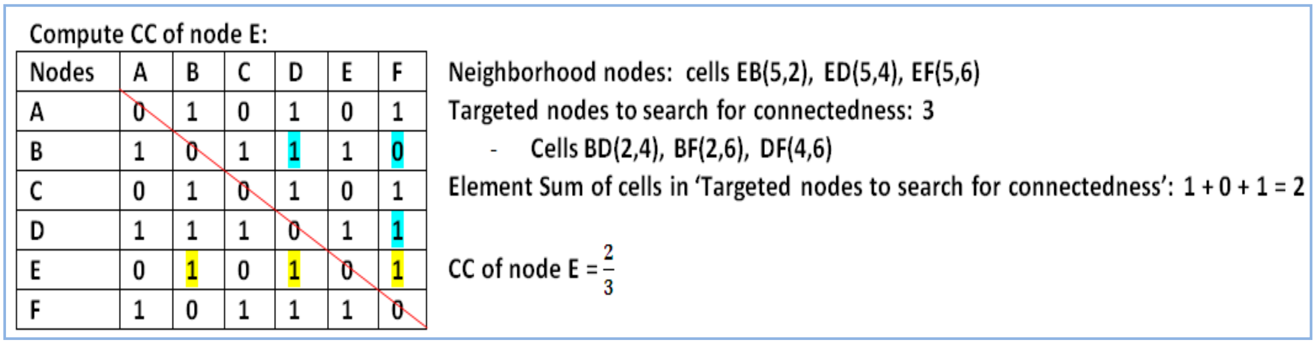
Targeted nodes to search for connectedness: 6

- Cells AB(1,2) AC(1,3), AE(1,5), AF(1,6), BC(2,3), BE(2,5), DF(2,6), CE(3,5), CF(3,6), EF(5,6)

Element Sum of cells in 'Targeted nodes to search for connectedness': 1 + 0 + 0 + 1 + 1 + 1 + 0 + 0 + 1 + 1 = 6

CC of node D = $\frac{6}{10} = 0.6$

Table 3.7 – CC Algorithm representation of node D



Compute CC of node E:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells EB(5,2), ED(5,4), EF(5,6)

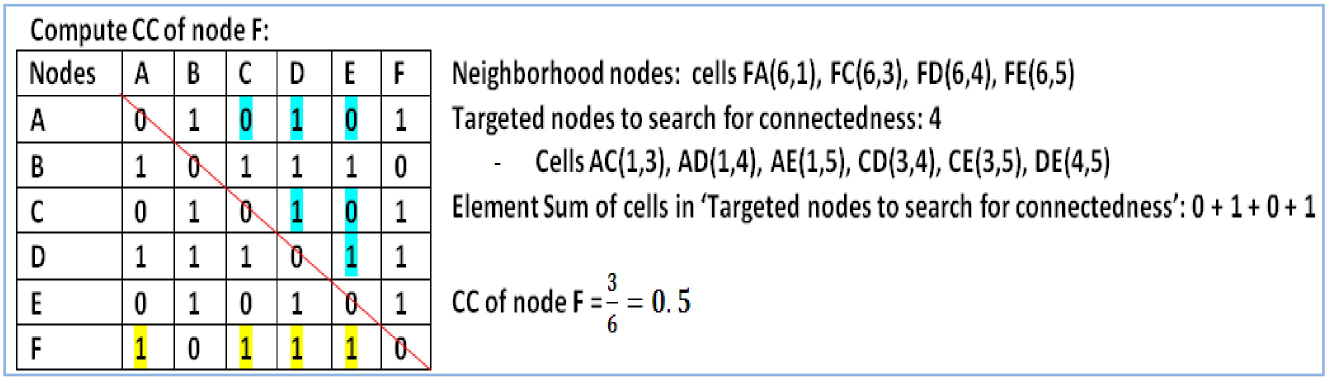
Targeted nodes to search for connectedness: 3

- Cells BD(2,4), BF(2,6), DF(4,6)

Element Sum of cells in 'Targeted nodes to search for connectedness': 1 + 0 + 1 = 2

CC of node E = $\frac{2}{3}$

Table 3.8 – CC Algorithm representation of node E



Compute CC of node F:

Nodes	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	1	1	1	0
C	0	1	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	1	0	1	1	1	0

Neighborhood nodes: cells FA(6,1), FC(6,3), FD(6,4), FE(6,5)

Targeted nodes to search for connectedness: 4

- Cells AC(1,3), AD(1,4), AE(1,5), CD(3,4), CE(3,5), DE(4,5)

Element Sum of cells in 'Targeted nodes to search for connectedness': 0 + 1 + 0 + 1

CC of node F = $\frac{3}{6} = 0.5$

Table 3.9 – CC Algorithm representation of node F

From the above tables, there are a few features of this method which is not only interesting but able to bring out the characteristics of network graphs. First, take the computation of CC of A for example, observe the neighborhood node cells of node A are AB(1,2), AD(1,4) and AF(1,6). At this point notice the column cells of the three nodes, which is 2, 4 and 6. Given this column cells numbers that were picked out, a pattern can be seen in the following statement which is the targeted nodes cells that were chosen based on the CC method adopted. That is, the cells BD(2,4), BF(2,6) and DF(4,6) only came from the numbers 2, 4 and 6 that was taken from the initial neighborhood nodes column cells. Furthermore, notice the condition of the cells selected works only when y>x in (x,y). With the observations of these few features for the CC computation, the CC can be accurately calculated by taking the element sum of the cells that were chosen. Note that the element sum only involves two values "1" and "0". This is in conjunction with the fact that the element sum only involves direct edge connection between the neighborhood nodes of the subject node.

3.3 Degree Distribution

3.3.1 Degree Distribution Derivation

Previously defined in Chapter 2.1.2, the degree of a node is in fact the number of neighboring nodes the subject node has direct edge connection with. The degree distribution of a network is thus defined to be the fraction of nodes in the network with degree k. That is, if there are n nodes in total in the network and n_k of them have degree k, in mathematical terms, $P(k) = \frac{n_k}{n}$.

3.3.2 Degree Distribution Data Computation

To compute the degree distribution of a network, the method of array cells is once again used in the DGD Algorithm. Similar to the computing of CC Algorithm, the DGD Algorithm makes use of direct edge connections to first compute the degree of the particular node in the network. After which, there was a need to calculate the frequency of the various degrees available in the network. This agrees with the theoretical derivation in Chapter 3.3.1 where n_k counts the total number of nodes with the same degree k. Finally, the degree distribution is calculated by summing all degrees available over the total number of nodes in the network.

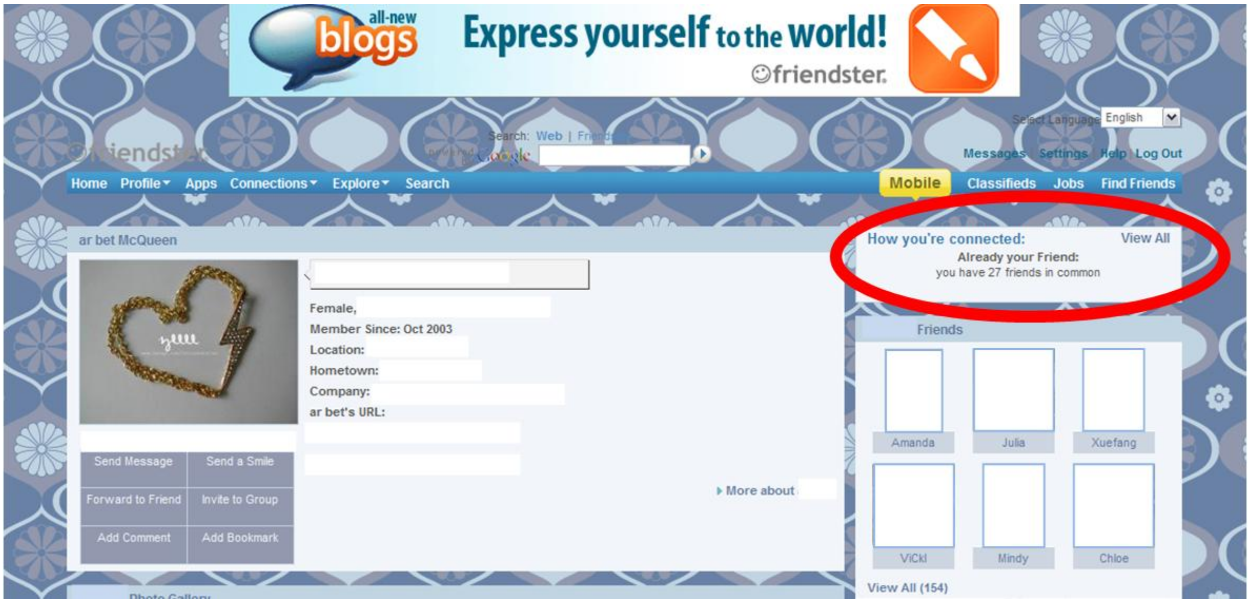
3.3.3 Degree Distribution Algorithm

Follow up with Chapter 3.3.2, the DGD Algorithm is designed such that the histogram plotting method will be adopted to calculate the frequency of all available degrees in the network. By plotting in the form of a histogram, the available degrees in the particular network will be counted and displayed accordingly in ascending order before the graph is displayed to check whether it follows a Poisson distribution or the Power Law tail. More results for DGD will therefore be discussed in Chapter 4.

Chapter 4 Experimental Results

4.1 Friendster Social Network

Before the Friendster Network is investigated, the cell array of the Friendster Network was taken manually by studying the common “friends” that a particular friend has in relation to the subjected friend “Vic”, such that “Vic” was taken to be the main connecting “friend” of the network that have direct edge connection to the corresponding 203 friends. In this context, the friends will be taken as the nodes of the network and the relationship of knowing and connecting to each other will be the edges. However, the main node “Vic” is not considered in the computation for the network properties to prevent biasness of the network to node “Vic”. Thus, before applying the 3 Algorithms on this network, the cell array of this network have to be first tabulated. For such a virtual Social Network on the Internet, an approach adopted was to compare the common “friends” that “Vic” has to the other nodes one by one to obtain the cell array of data in the first form, that is to have only the data of direct edge connections and no direct connections, indicated by “1”s and “0”s in the cell array. Figure 4.1 shows the format to which the Friendster Social Network is being displayed. A sample of the data computation and excel file is shown in Figure 4.2 and Table 4.1 respectively.

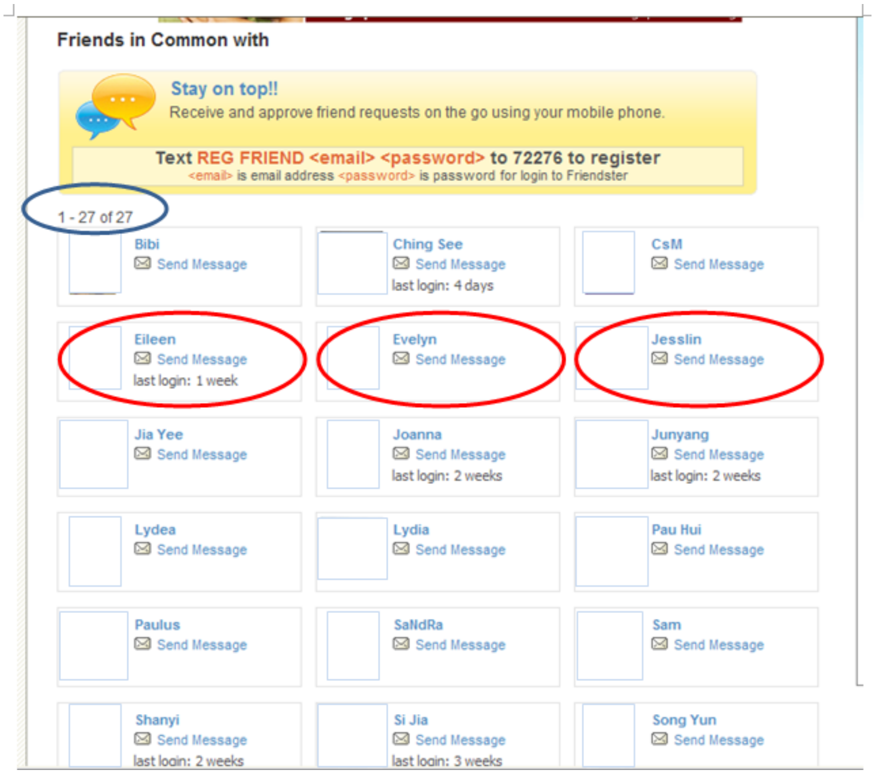


ar bet McQueen

Female,
Member Since: Oct 2003
Location:
Hometown:
Company:
ar bet's URL:

How you're connected:
Already your Friend:
you have 27 friends in common

Figure 4.1 – Sample of Friendster Social Network on the Internet. Common “friends” (nodes) between a particular friend with the main friend (nodes) are displayed in the red circle above.



Friends in Common with

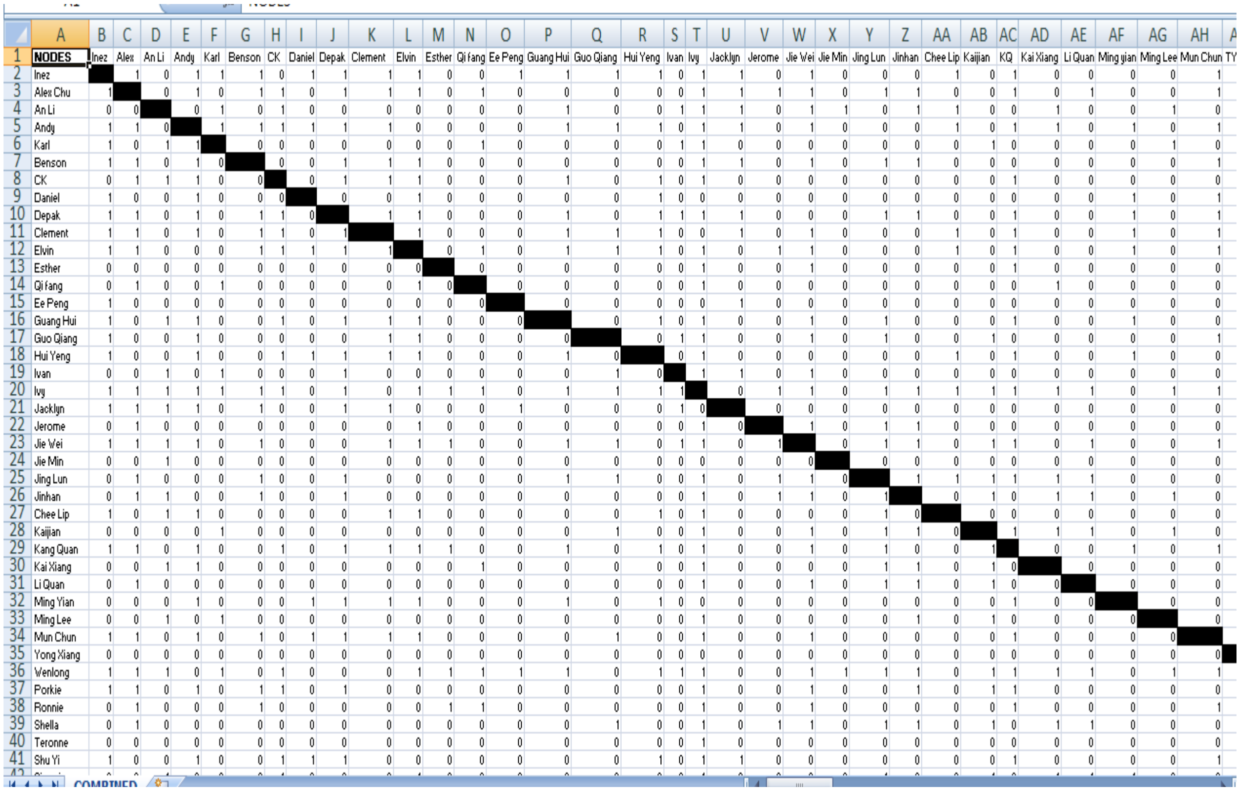
Stay on top!!
Receive and approve friend requests on the go using your mobile phone.

Text REG FRIEND <email> <password> to 72276 to register
<email> is email address <password> is password for login to Friendster

1 - 27 of 27

Bibi	Ching See last login: 4 days	CsM
Eileen last login: 1 week	Evelyn	Jesslin
Jia Yee	Joanna last login: 2 weeks	Junyang last login: 2 weeks
Lydea	Lydia	Pau Hui
Paulus	SaNdRa	Sam
Shanyi last login: 2 weeks	Si Jia last login: 3 weeks	Song Yun

Figure 4.2 – Sample of Data computation of common nodes from Friendster network. Common nodes that both the main node and other node have are in red circle. Each circle denotes one common node. Blue circle indicate the number of common nodes that both nodes have together.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	A
1	**NODES**	Inez	Alex	An Li	Andy	Karl	Benson	CK	Daniel	Depak	Clement	Elvin	Esther	Qi fang	Ee Peng	Guang Hui	Guo Qiang	Hui Yeng	Ivan	Ivy	Jacklyn	Jerome	Jie Wei	Jie Min	Jing Lun	Jinhan	Chee Lip	Kaijian	KQ	Kai Xiang	Li Quan	Ming yian	Ming Lee	Mun Chun	TY
2	Inez	■	1	0	1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	
3	Alex Chu	1	■	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	1	0	0	1	0	1	0	0	1	
4	An Li	0	0	■	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	
5	Andy	1	1	0	■	1	1	1	1	1	1	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	1	0	1	1	0	1	0	1	
6	Karl	1	0	1	1	■	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	
7	Benson	1	1	0	1	0	■	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	
8	CK	0	1	1	1	0	0	■	0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
9	Daniel	1	0	0	1	0	0	0	■	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
10	Depak	1	1	0	1	0	1	1	0	■	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0	1	
11	Clement	1	1	0	1	0	1	1	0	1	■	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	
12	Elvin	1	1	0	0	0	1	1	1	1	1	■	0	1	0	1	1	1	0	1	0	1	1	0	0	0	1	0	1	0	0	1	0	1	
13	Esther	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	
14	Qi fang	0	1	0	0	1	0	0	0	0	0	1	0	■	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
15	Ee Peng	1	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	Guang Hui	1	0	1	1	0	0	1	0	1	1	1	0	0	0	■	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	
17	Guo Qiang	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	■	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	
18	Hui Yeng	1	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0	■	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	0	
19	Ivan	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	■	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	
20	Ivy	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	■	0	1	1	0	1	1	1	1	1	1	1	0	1	1	
21	Jacklyn	1	1	1	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	Jerome	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	■	1	0	1	1	0	0	0	0	0	0	0	0	
23	Jie Wei	1	1	1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	1	1	0	1	■	0	1	1	0	1	1	0	1	0	0	1	
24	Jie Min	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	
25	Jing Lun	0	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	■	1	1	1	1	1	1	0	0	0	
26	Jinhan	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	■	0	1	0	1	1	0	1	0	
27	Chee Lip	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	■	0	0	0	0	0	0	0	
28	Kaijian	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	0	■	1	1	1	0	1	0	
29	Kang Quan	1	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1	■	0	0	1	0	1	
30	Kai Xiang	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	■	0	0	0	0	
31	Li Quan	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	1	0	0	■	0	0	0	
32	Ming Yian	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	■	0	0	
33	Ming Lee	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	■	0	
34	Mun Chun	1	1	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	■	
35	Yong Xiang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■
36	Wenlong	1	1	1	0	1	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	1	
37	Porkie	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0	
38	Ronnie	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	1	
39	Shella	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	1	1	0	1	0	1	1	0	0	0	
40	Teronne	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
41	Shu Yi	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	

Table 4.1 – Somple of part of Cell array of Friendster Network computed in excel file. Table is computed with data for nodes with direct edge connections. Full Table for Network can be found in Appendix.

4.2 Average Path Length of Friendster Network

Results were taken from the APL Algorithm created as listed below:

Total nodes in Friendster Network = 203 nodes

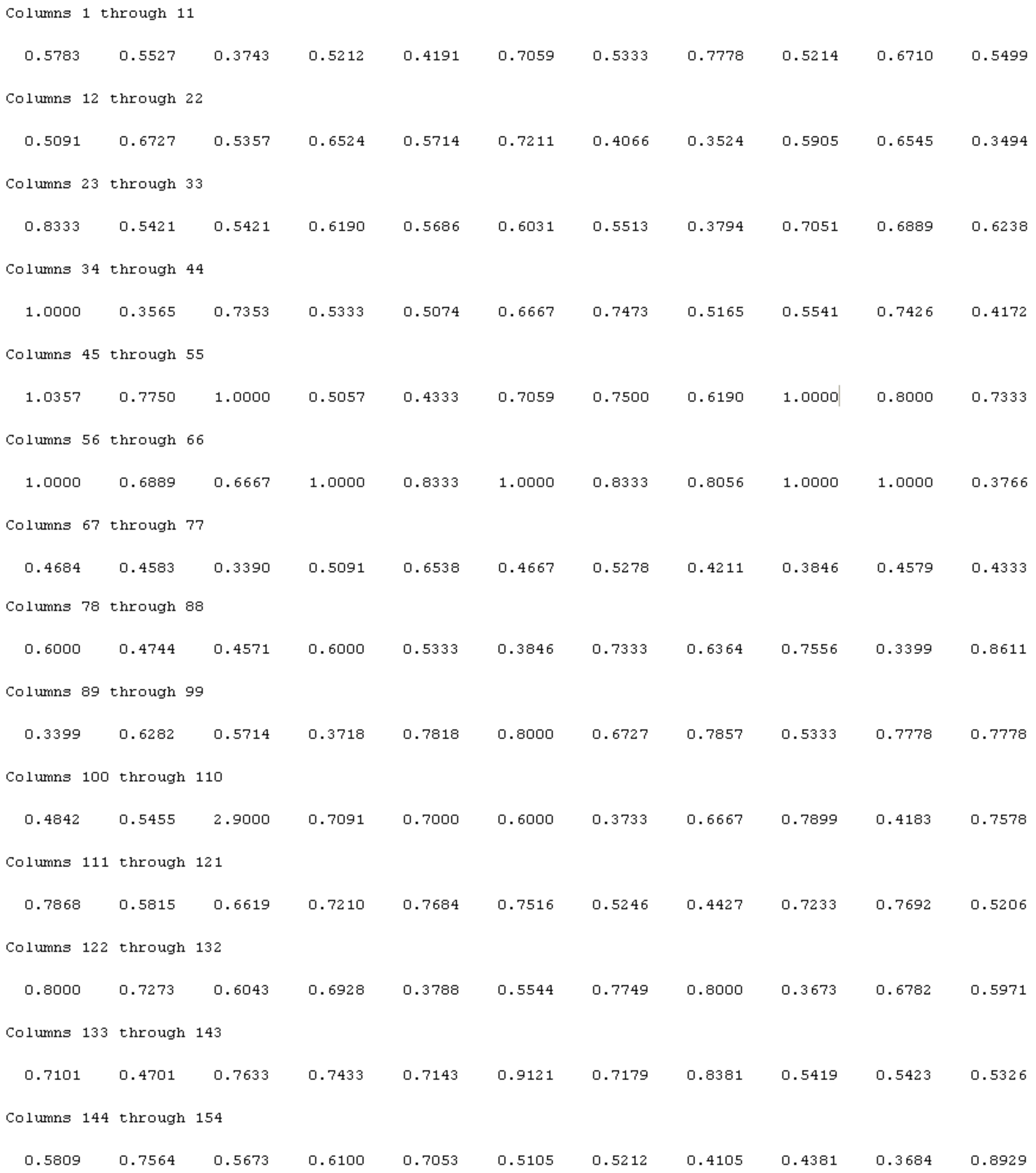
Maximum number of edges possible = 20706

Total number of shortest paths edges in Friendster Network = 61622

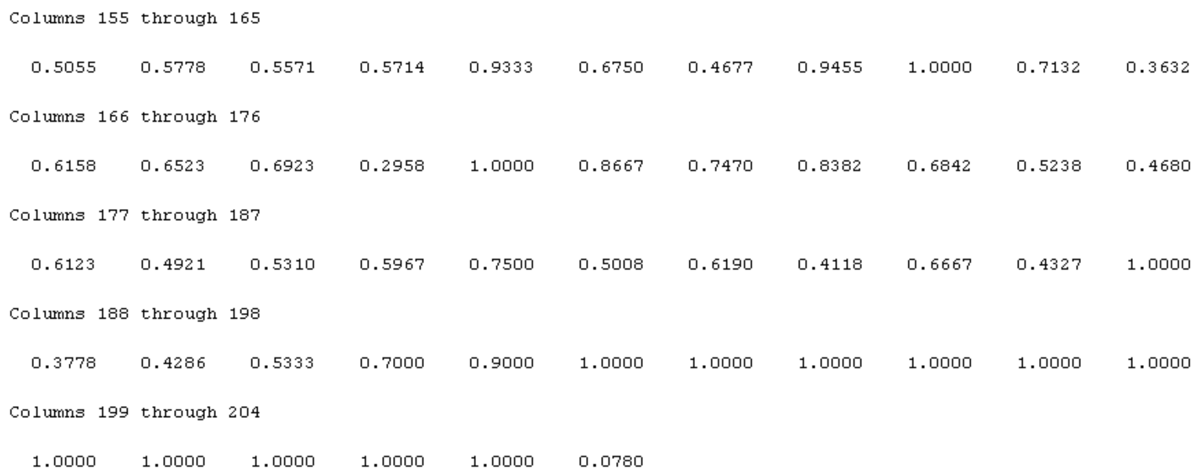
Average Path Length of Friendster Network = 2.9956

Results for APL shown above were taken from the MATlab program after running the APL Algorithm. Total number of shortest paths edges reflects the total number of shortest paths that each individual node takes to travel to every other node in the network. The value was taken from the upper triangular array of the network to prevent double counting. After which, the APL is then calculated by the taking the sum of the total number of shortest paths edges over the total number of nodes in this network.

4.3 Clustering Coefficient of Friendster Network



```
  Columns 1 through 11

    0.5783    0.5527    0.3743    0.5212    0.4191    0.7059    0.5333    0.7778    0.5214    0.6710    0.5499

  Columns 12 through 22

    0.5091    0.6727    0.5357    0.6524    0.5714    0.7211    0.4066    0.3524    0.5905    0.6545    0.3494

  Columns 23 through 33

    0.8333    0.5421    0.5421    0.6190    0.5686    0.6031    0.5513    0.3794    0.7051    0.6889    0.6238

  Columns 34 through 44

    1.0000    0.3565    0.7353    0.5333    0.5074    0.6667    0.7473    0.5165    0.5541    0.7426    0.4172

  Columns 45 through 55

    1.0357    0.7750    1.0000    0.5057    0.4333    0.7059    0.7500    0.6190    1.0000    0.8000    0.7333

  Columns 56 through 66

    1.0000    0.6889    0.6667    1.0000    0.8333    1.0000    0.8333    0.8056    1.0000    1.0000    0.3766

  Columns 67 through 77

    0.4684    0.4583    0.3390    0.5091    0.6538    0.4667    0.5278    0.4211    0.3846    0.4579    0.4333

  Columns 78 through 88

    0.6000    0.4744    0.4571    0.6000    0.5333    0.3846    0.7333    0.6364    0.7556    0.3399    0.8611

  Columns 89 through 99

    0.3399    0.6282    0.5714    0.3718    0.7818    0.8000    0.6727    0.7857    0.5333    0.7778    0.7778

  Columns 100 through 110

    0.4842    0.5455    2.9000    0.7091    0.7000    0.6000    0.3733    0.6667    0.7899    0.4183    0.7578

  Columns 111 through 121

    0.7868    0.5815    0.6619    0.7210    0.7684    0.7516    0.5246    0.4427    0.7233    0.7692    0.5206

  Columns 122 through 132

    0.8000    0.7273    0.6043    0.6928    0.3788    0.5544    0.7749    0.8000    0.3673    0.6782    0.5971

  Columns 133 through 143

    0.7101    0.4701    0.7633    0.7433    0.7143    0.9121    0.7179    0.8381    0.5419    0.5423    0.5326

  Columns 144 through 154

    0.5809    0.7564    0.5673    0.6100    0.7053    0.5105    0.5212    0.4105    0.4381    0.3684    0.8929
```



```
  Columns 155 through 165

    0.5055    0.5778    0.5571    0.5714    0.9333    0.6750    0.4677    0.9455    1.0000    0.7132    0.3632

  Columns 166 through 176

    0.6158    0.6523    0.6923    0.2958    1.0000    0.8667    0.7470    0.8382    0.6842    0.5238    0.4680

  Columns 177 through 187

    0.6123    0.4921    0.5310    0.5967    0.7500    0.5008    0.6190    0.4118    0.6667    0.4327    1.0000

  Columns 188 through 198

    0.3778    0.4286    0.5333    0.7000    0.9000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000

  Columns 199 through 204

    1.0000    1.0000    1.0000    1.0000    1.0000    0.0780
```

Results taken from CC Algorithm shows:

Clustering Coefficient of Friendster Network = 0.5927

4. Degree Distribution of Friendster Network

Results for the Degree Distribution, P(k) of the Friendster Network are as follows in Table 4.2:

Degree, k	Degree Distribution, P(k)	Degree, k	Degree Distribution, P(k)
2	0.0147	22	0.0196
3	0.0245	23	0.0147
4	0.0588	24	0.0196
5	0.0343	25	0.0245
6	0.0343	26	0.0147
7	0.0098	27	0.0392
8	0.0539	28	0.0245
9	0.0343	29	0.0049
10	0.0441	30	0.0147
11	0.0539	31	0.0098
12	0.0049	32	0.0049
13	0.0441	33	0.0049
14	0.0441	34	0.0196
15	0.0294	35	0.0147
16	0.0294	38	0.0147
17	0.0441	41	0.0147
18	0.0343	50	0.0049
19	0.0245	53	0.0049

20	0.0686		
21	0.0441		

Table 4.2 – Degree Distribution results from Friendster Network

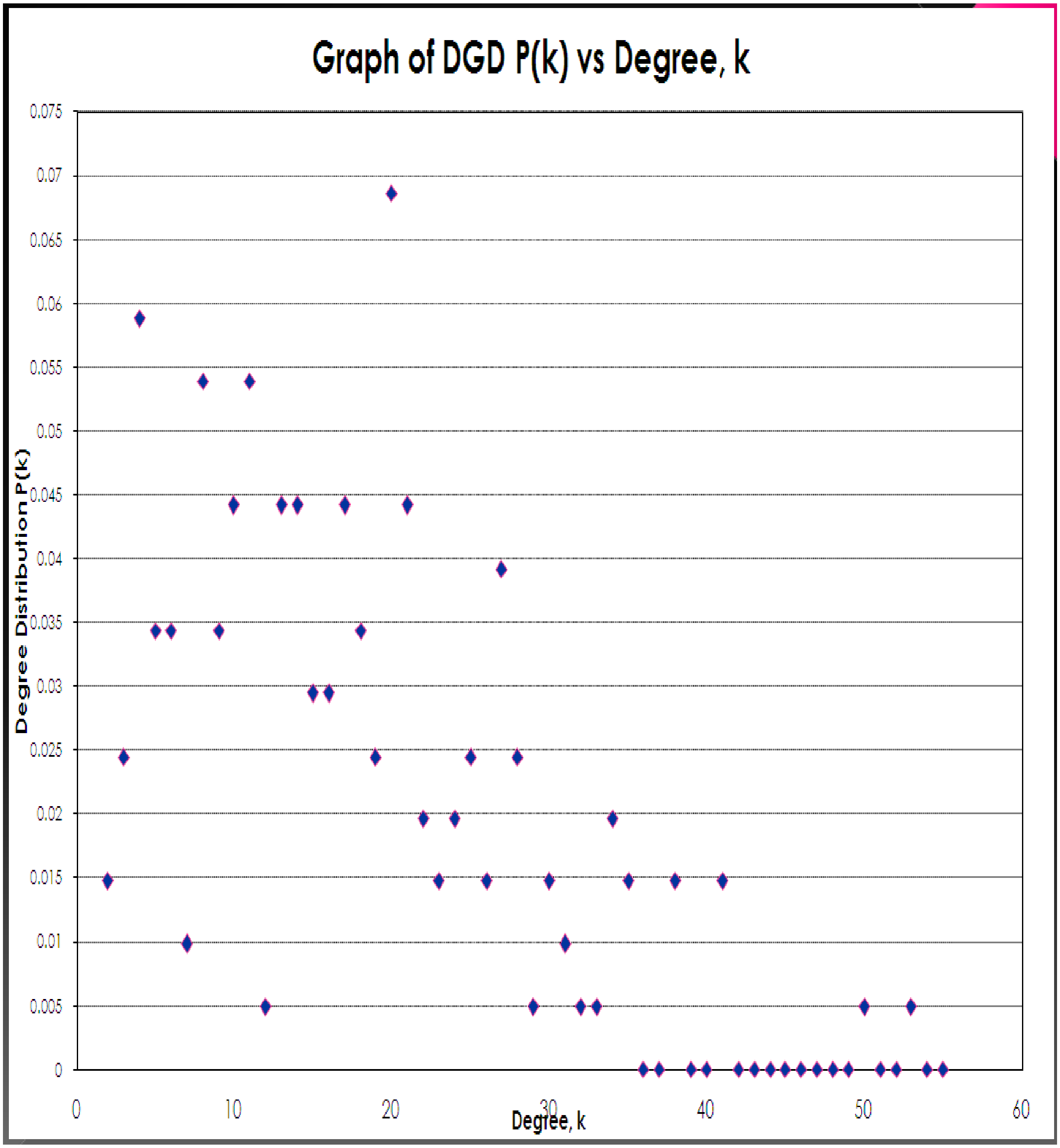


Figure 4.3 – Graph of Degree Distribution, P(k) vs Degree, k for Friendster Network

Chapter 5 Analysis and Discussion

5.1 Average Path Length

Quoted from "Statistical Physics of complex networks" by Réka Albert and Albert-László Barabási, Table 5.1 shows the various Average Path Lengths of several networks that were previously worked upon. Note that not all networks were taken into comparison due to the size of the network sample being tested.

Network	Size	APL	APL_{random}	Reference
Ythan Estuary Food Web	134	2.43	2.26	Montoya and Solé, 2000
Silwood Park Food Web	154	3.40	3.23	Montoya and Solé, 2000
C. Elegans	282	2.65	2.25	Watts and Strogatz, 1998
E. coli, substrate graph	282	2.9	3.04	Wagner and Fell, 2000
E. coli, reaction graph	315	2.62	1.98	Wagner and Fell, 2000
Friendster Social Network	***203***	***2.9956***	***1.851***	***N.A***

Table 5.1 – General Characteristics of Average Path Length of real networks with the Friendster Network.

The networks used for comparisons were chosen based on node size in order to provide a similar comparison between the systems and networks in terms of Average Path Lengths. In addition, the Average Path Lengths are also compared to the Average Path Length for random graphs of the same size and average degree <k>, APL_{random}, for further comparisons if needed. Based on the results in Table 5.1, it can be suggested that the Friendster Social Network exhibits a similar low average path length as several APL of real life networks exhibit, which is also said to be close to APL of Random Graphs. In addition, a point to be taken note is the range of values for small or low APL. As recorded in several literature surveys (10), the values for low average path lengths to be taken for this project would be values that are 1<APL<~4. As such, a few examples of low APL are listed in Table 5.1. On the other hand, values of large APLs can be taken from examples of the study of WWW, where it has been found that the APL of WWW was found to be 16 and 19 at two different points of research. (10)

5.2 Clustering Coefficient

We compare the results for the clustering coefficients of the Friendster Social Network with the same real life networks taken in Table 5.1 as shown below:

Network	**Size**	**CC**	CC_{random}	**Reference**
Ythan Estuary Food Web	134	0.22	0.06	Montoya and Solé, 2000
Silwood Park Food Web	154	0.15	0.03	Montoya and Solé, 2000
C. Elegans	282	0.28	0.05	Watts and Strogatz, 1998
Escherichia coli, substrate graph	282	0.32	0.026	Wagner and Fell, 2000
Escherichia coli, reaction graph	315	0.59	0.09	Wagner and Fell, 2000
Friendster Social Network	***203***	***0.5927***	***0.0867***	***N.A***

Table 5.2 - General Characteristics of Clustering Coefficients of real networks with the Friendster Network.

Base on theories studied in Chapter 2, Small-World Networks are proven to have unusually large clustering coefficients which is similar to food webs shown by Montoya and Solé, where food webs are highly clustered as well. Furthermore, Wagner and Fell had previously studied the *Escherichia coli* bacterium graphs to determine that the undirected versions of these graphs prove to have a large clustering coefficient. With these theories in place, it can be suggested that the Friendster Social Network thus exhibit similar large clustering coefficient feature as in the other real life networks in comparison. As such, the Friendster Social Network can also be said to not follow the CC feature of a random graph due to the large difference of 658.131% between CC and CC_{random} as in Table 5.2. A point to take note in this comparison is that the networks being compared are also found to be in the undirected edges version, which was previously discussed in Chapter 3.1.2, which makes the comparison more accurate and meaningful for comparison purposes. As such, the Friendster Social Network appears to exhibit the properties of Small-World Networks in terms of possessing large Clustering Coefficients. Also, the values of CC are taken to be in the range where small CC are classified to be lower than 0.09 and large CC can be in the range of more than 0.1. (10) As such, the reliability and stability of the values obtained for both APL and CC can be justified and verified for research purposes.

5.3 Degree Distribution and Power Law Tail

Network	Size	α	Reference
Ythan Estuary Food Web	134	1.05	Montoya and Solé, 2000
Silwood Park Food Web	154	1.13	Montoya and Solé, 2000
Friendster Social Network	***203***	***0.9803***	***N.A***

Table 5.3 – General Characteristics of exponent factor of real networks with Friendster Network

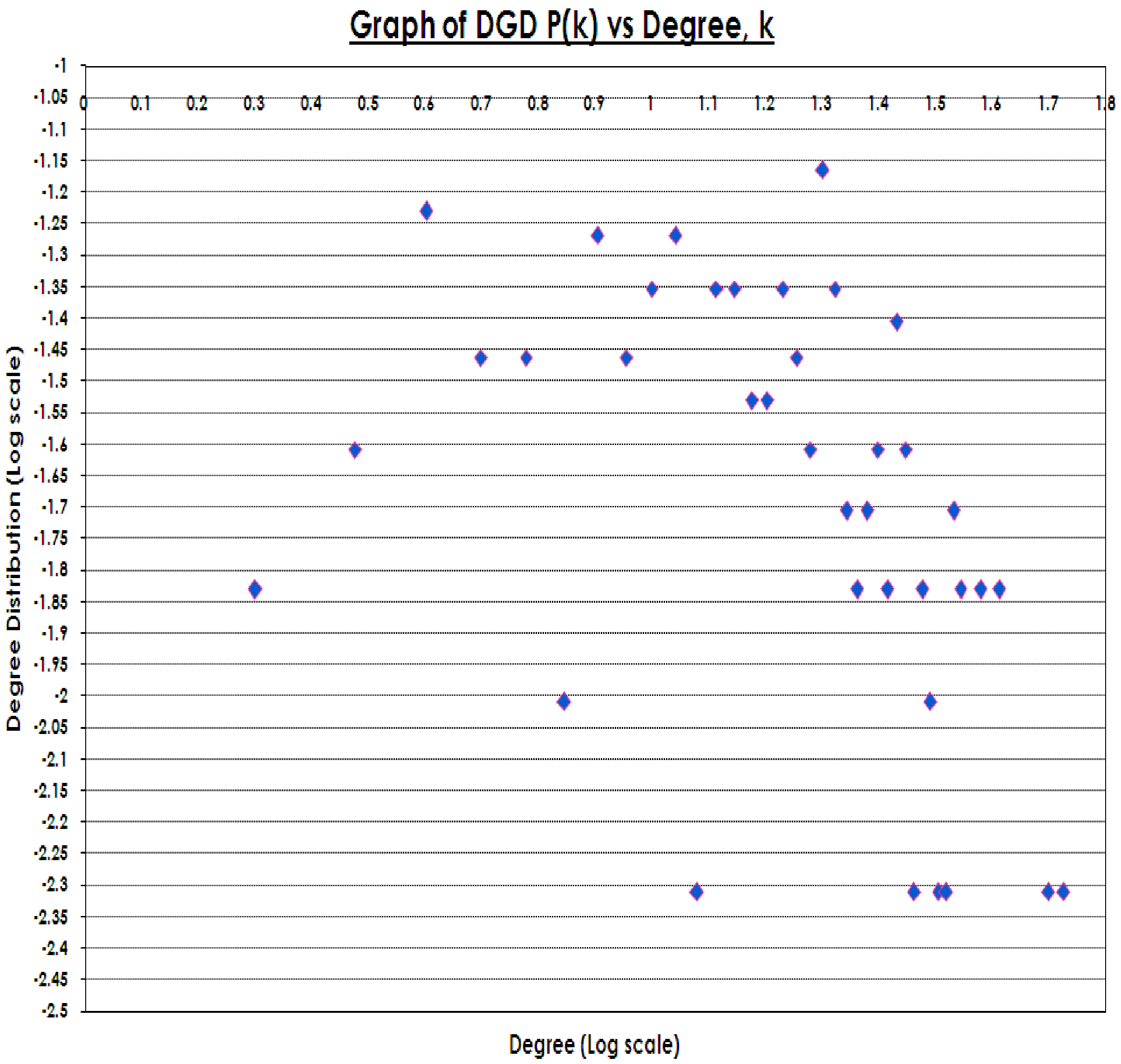


Figure 5.1 – Power Law Plot of Friendster Social Network.

As shown in Figure 5.1, the Power Law Distribution appears to be in negative y-axis as shown. To determine the power law tail and exponent value, the absolute values of the y-axis is taken and plotted in Figure 5.1(a).

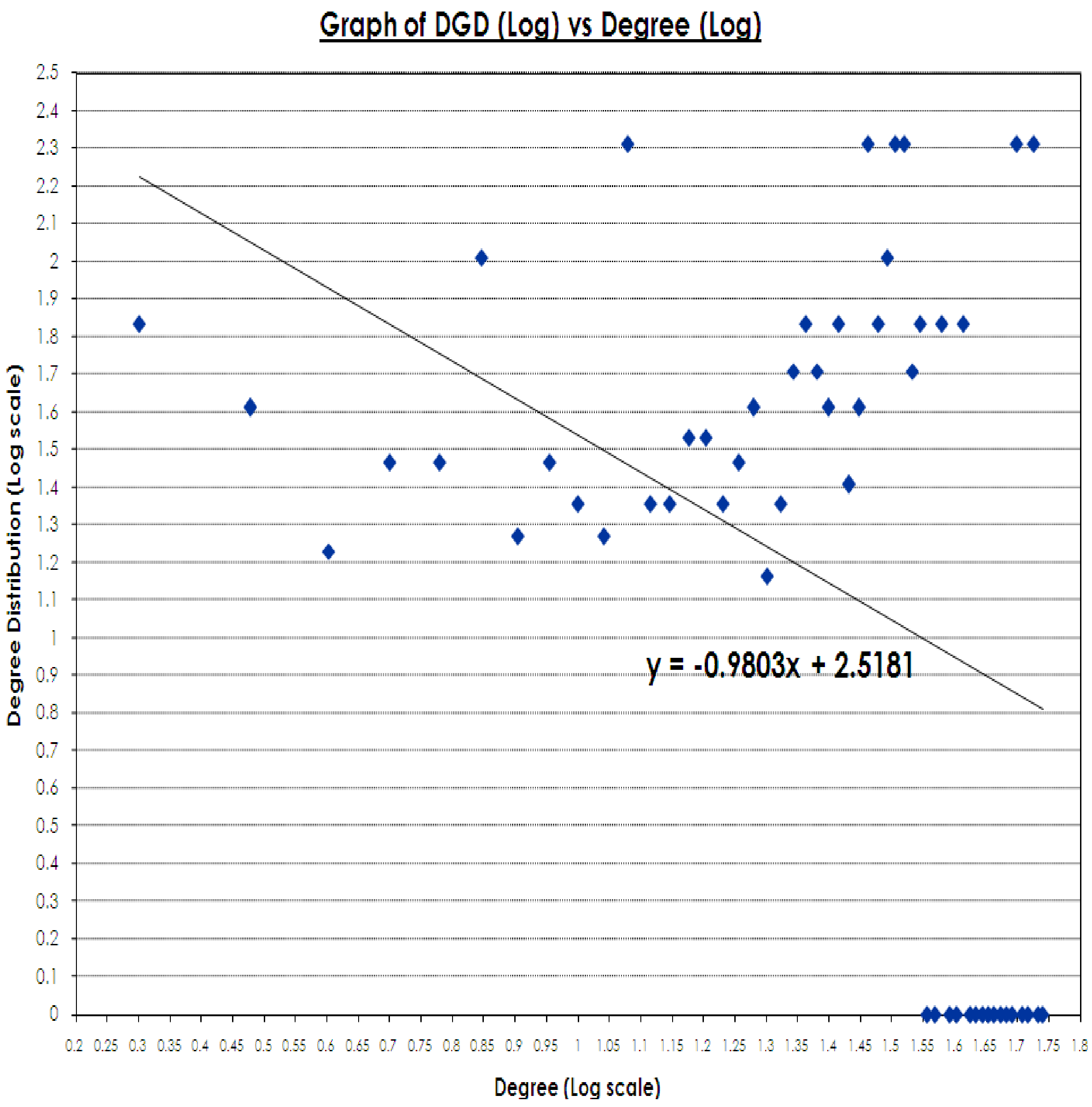


Figure 5.1(a) – Power Law Distribution Graph of Friendster Social Network

5.3.1 Analysis of Removal of Nodes

With the values as computed previously, the next question to ask would be the stability and biasness of this network. As such, another consideration would be to remove random nodes from the network to carry on analysis. As introduced by Mark Newman in "The Physics of networks", suppose that some nodes in the network were removed for some reasons, it does not really affect the network to a large extend due to the fact in most of the networks, there are always more than one alternative routes and edges to the other points. Therefore, as mentioned by Réka Albert and Albert-László Barabási (10), they found that it depends on the degree distribution to a large extend.

These researchers had actually considered to methods of removing nodes, which will be carried out in this part of the analysis to make further comparisons. The first method was to remove the highest-degree node in the network and the other was to remove nodes at random. They had initially assumed the first method would lead to the network failing faster, but in fact little difference was observed between two methods. In the following analysis, Figure 5.2 and Figure 5.2(a) reflects the updated Graph of Degree Distribution, P(k) vs Degree, k after the removal of the highest-degree node, which is the node 53 in the Friendster Network and its subsequent Power Law Distribution Graph respectively. Figure 5.3 and Figure 5.3(a) reflects the updated Graph of Degree Distribution, P(k) vs Degree, k after random removal of 5 nodes and its subsequent Power Law Distribution Graph respectively.

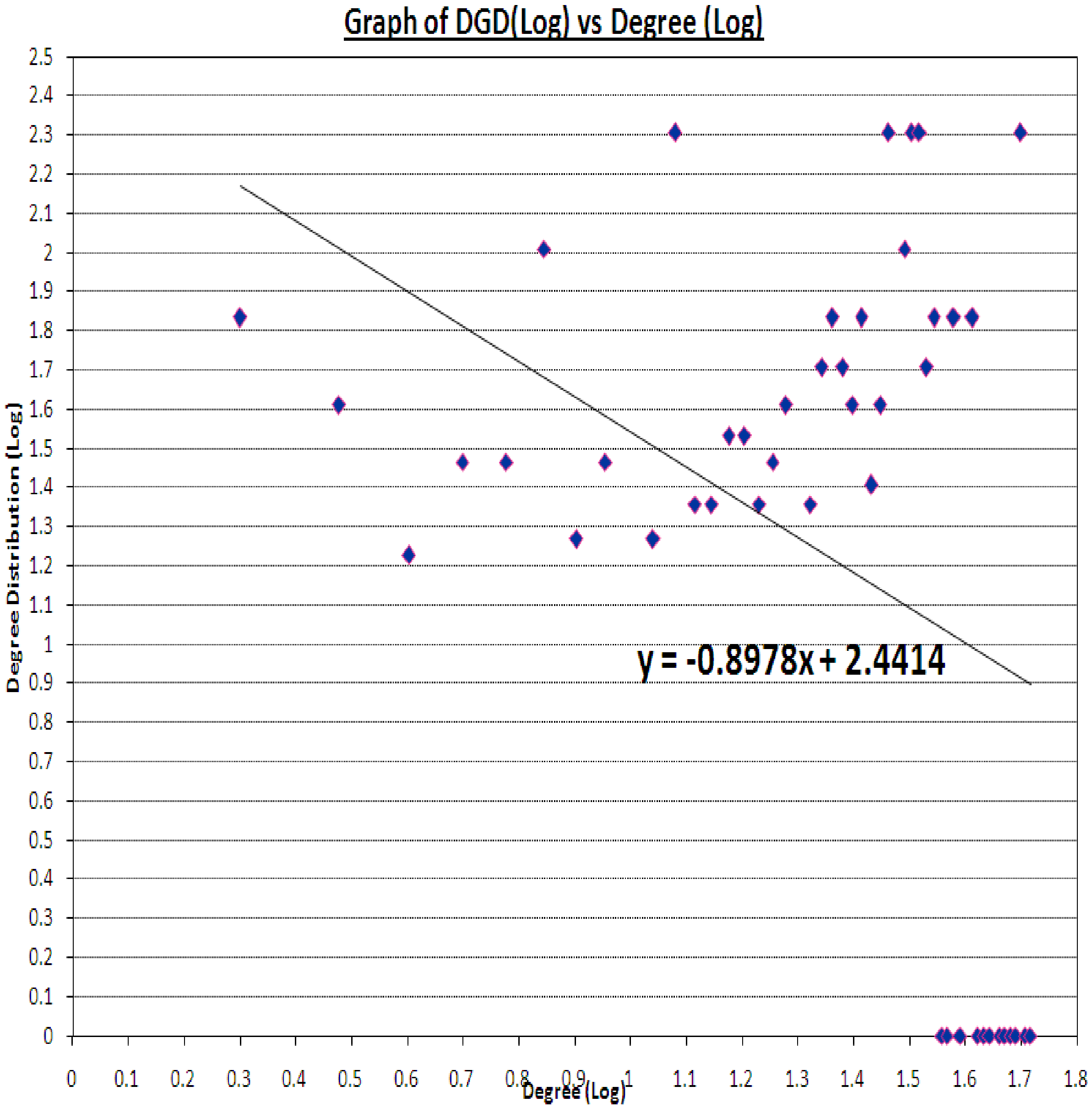


Figure 5.2 – Power Law Distribution Graph of Friendster Network after first method removal of node

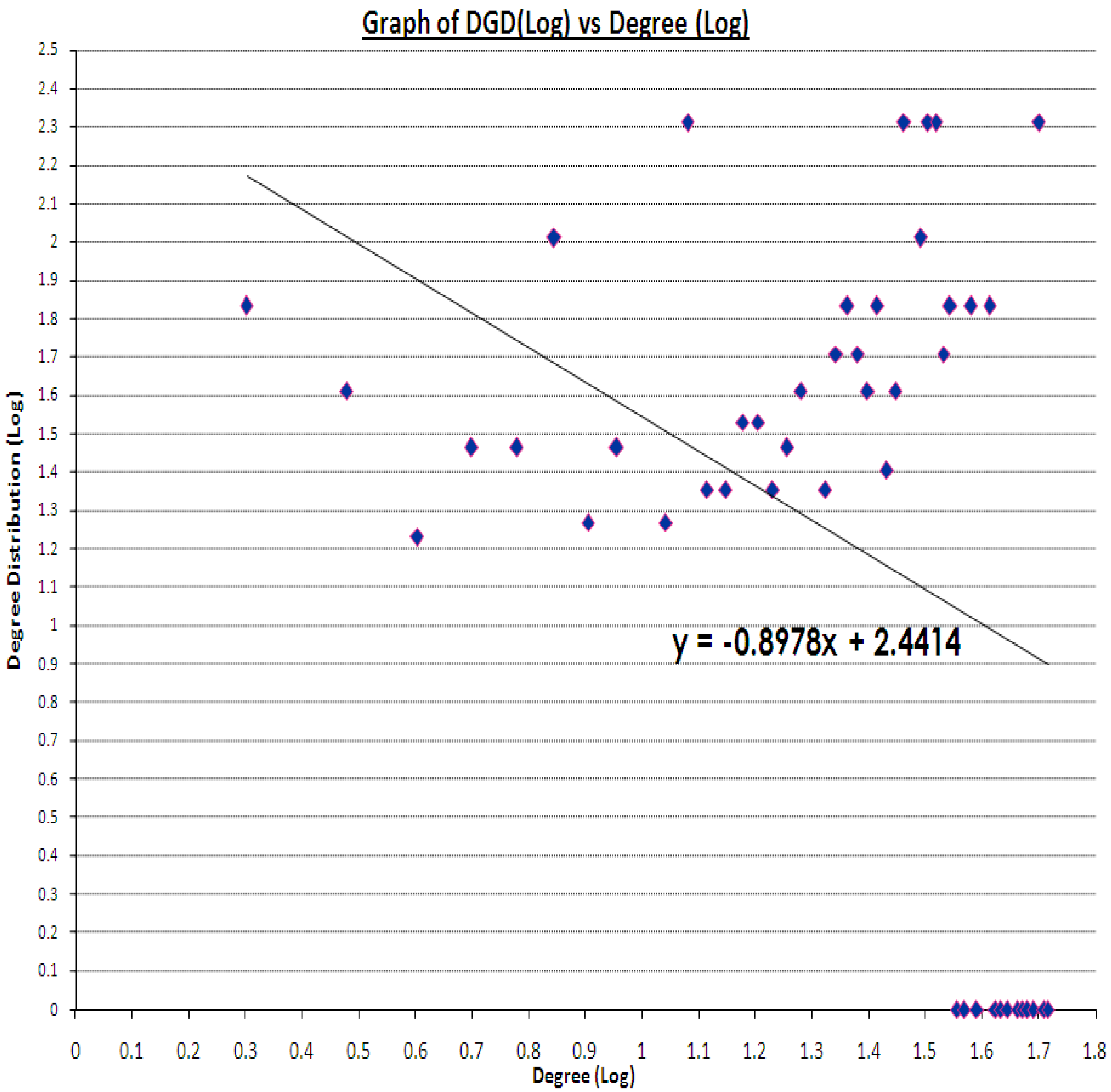


Figure 5.3 - Power Law Distribution Graph of Friendster Network after second method removal of 5 random nodes

To compare the power law distribution exponent, first consider the graph plotted in the logarithmic scale, where$y = Cx^{-a}$, where a is the exponent. The logarithmic trend lines will be obtained in the form of $\log(y) = \log(C) - alog(x)$. As such, Comparing the coefficients of a, the exponent of the Friendster Network are obtained as follows. Notice from Figures 5.1(a), 5.2 and 5.3, the exponents obtained are:

Original Friendster Network = **0.9803**

Friendster Network with removal of Highest Degree Node = **0.8978**

Friendster Network with random removal of 5 nodes = **0.8978**

The minute difference between the exponents thus correspond to the theoretical point that was made earlier on by Mark Newman in this physics article with respect to the two methods of removal of nodes. Moreover, in comparison with real life networks reflected in Table 5.3, the values of the Friendster Network exponent also corresponds to the Real life networks exponent values in terms of the range of values. In conclusion, the Virtual Friendster Social Network appears to possibly agree with several network properties in terms of low APL and high CC of a Small-World Network, as well as the DGD, which produces a power-law tail distribution reflected in the graphs. As such, it can be suggested that the Friendster Network is a scale-free network and that, a Virtual Social Network may exhibit the similar characteristics as a real-life social network.

5.4 Other Considerations

With the Virtual Social Network being constructed and studied, a few considerations have to be noted to address the biasness and reliability of this network being researched on. First would be the age consideration. For all virtual social networks on the WWW, age would be a great affecting factor such that the main node (person being researched) that is being researched for its connecting nodes (friends) would be around the age group of the main node due to the environment that the main node is situated in, such as the school environment, where the main node will associate with friends of the same age group. Also, in the Singapore students' context, a general trend between the different main nodes would be that the connecting nodes to the main node will be limited by nodes of Primary School Friends Category, Secondary School Friends Category, Junior College or Polytechnic Friends Category as well as University Friends or other friends Category. As such, the virtual social network that was being researched on in this paper belongs to the Singapore Student Network Category.

5.5 Conclusion and Future Works

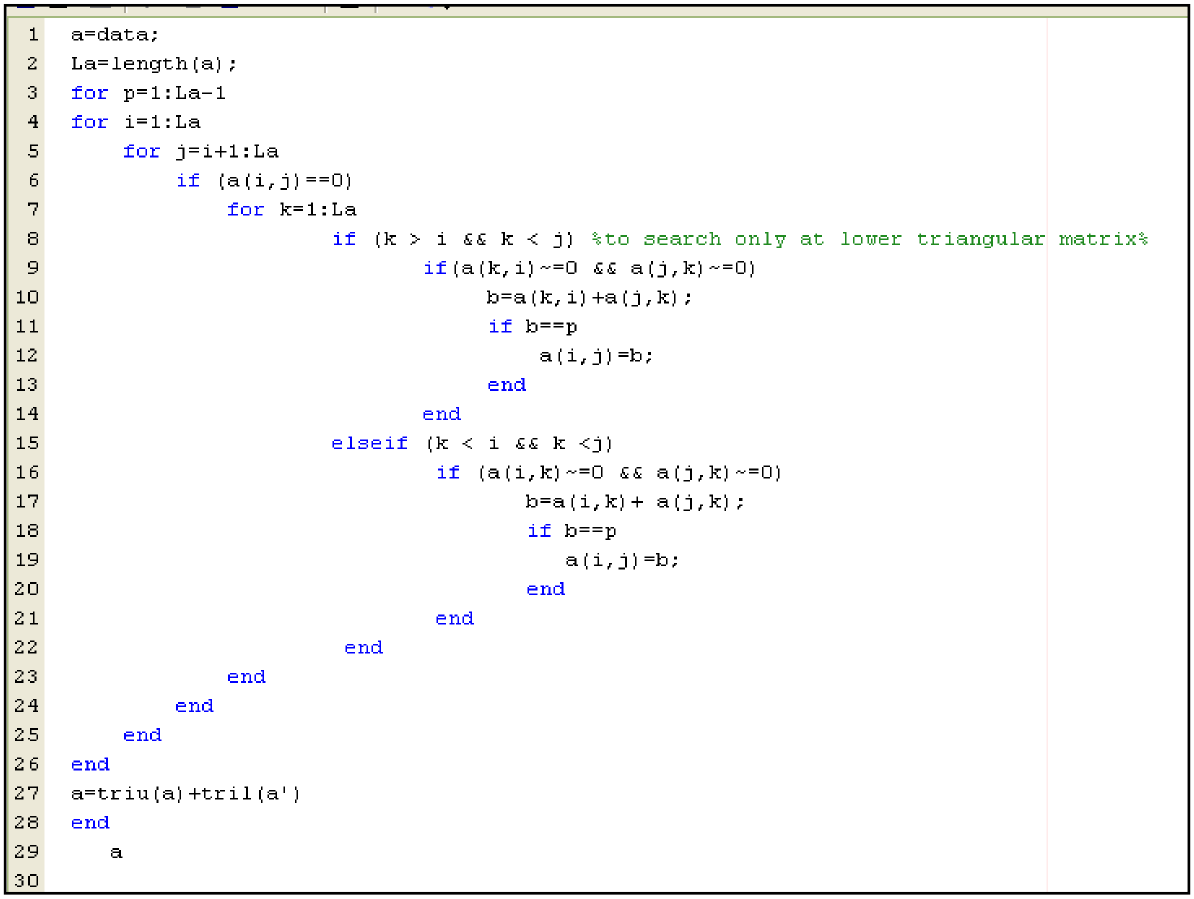
Up to this point, there are still many ongoing researches in network theory, be it on networks that have been debated many times or evolving new networks. There are several areas in the study of networks theories which have yet to be discovered or further researched into. With the increasing research and analysis in the physicists' perspective led to the discovery of several Network Graph Models as well as interesting properties of complex Network. One such evolution would therefore be the increasing Social networks that evolve from the increasing awareness of Internet, leading to more and more virtual social networking websites being created and researched on, such as Friendster, Facebook, Habbo and College Tonight etc, catering to networkings of different categories and age groups. As such, the research on the Friendster is therefore the stepping stone to further analysis and research on this ongoing expansion of these virtual social networks in the studies of Statistical Physics aspect of complex network theories.

References

1. Average Path Length. Retrieved on July 2008, from Wikipedia http://en.wikipedia.org/wiki/Average_path_length
2. Clustering Coefficient. Retrieved on January 2009. From Wandora, http://www.wandora.org/wandora/wiki/index.php?title=Clustering-coefficient
3. D.J Watts and Steven Strogatz (June 1998). "Collective dynamics of 'small-world' networks". Nature 393: 440-442.
4. Degree Distribution, Retrieved on July 2008. From Wikipedia http://en.wikipedia.org/wiki/Degree distribution
5. Graph Theory. Retrieved on September 2008. From Wikipedia http://en.wikipedia.org/wiki/Graph_theory
6. Mark Newman (2008) *Physics Today; The physics of networks. November Issue*. Retrieved on January 2009 from American Institute of Physics. Website: www.physicstoday.org
7. Pegg, Ed Jr. "Small World Network." From *MathWorld*--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/SmallWorldNetwork.html
8. Power Law. Retrieved on March 2009, from Wikipedia http://en.wikipedia.org/wiki/Power_Law.
9. Random Graph. Retrieved on December 2008. From Wikipedia http://en.wikipedia.org/wiki/Random_graph
10. Réka Albert and Albert-László Barabási (2002) *Review of Modern Physics; Statistical mechanics of complex networks, 74, P47-68.* Retrieved on June 2008 from The American Physical Society.
11. Scale Free Network. Retrieved on February 2009. From Wikipedia http://en.wikipedia.org/wiki/scale_free_network
12. Small-World Network. Retrieved on February 2009. From Wikipedia http://en.wikipedia.org/wiki/small-world_network

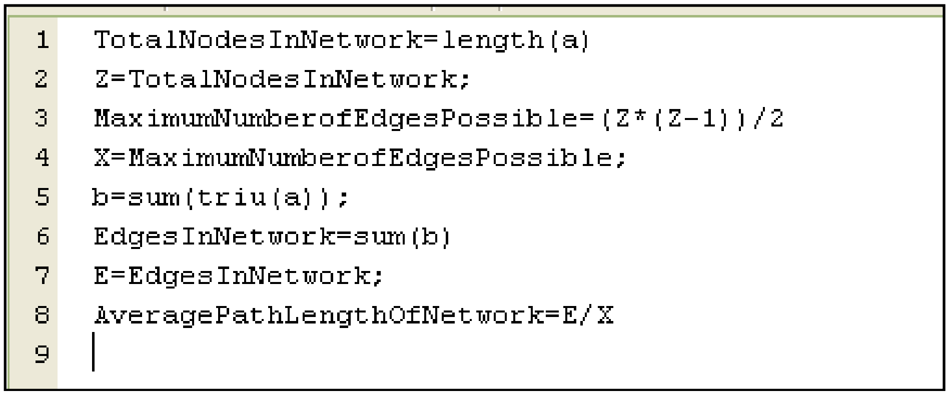
13. Weisstein, Eric W. "Random Graph." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/RandomGraph.html
14. Watts-Strogatz Model. Retrieved on March 2009. From Wikipedia http://en.wikipedia.org/wiki/Watts_and_Strogatz_model
15. Barthelemy, M.; Amaral, LAN. (1999). "Small-world networks: Evidence for a crossover picture". *Phys. Rev. Lett.* **82**: 3180
16. Balázs Szendröi, "The Structure of a large social network". Retrieved on February 2009
17. PJ Lamberson, "Networks". Retrieved on April 2009
18. Dorogovtsev, S.; Mendes, J. F. F. (2002). "Evolution of networks". *Advances in Physics* **51**: 1079–1187
19. Newman, M. E. J. (2003). "The structure and function of complex networks". *SIAM Review* **45**: 167–256.
20. Newman, Mark (2003). "The Structure and Function of Complex Networks". *SIAM Review* **45**: 167–256
21. Milgram, Stanley (1967). "The Small World Problem". *Psychology Today* **1** (1): 60–67

Appendix



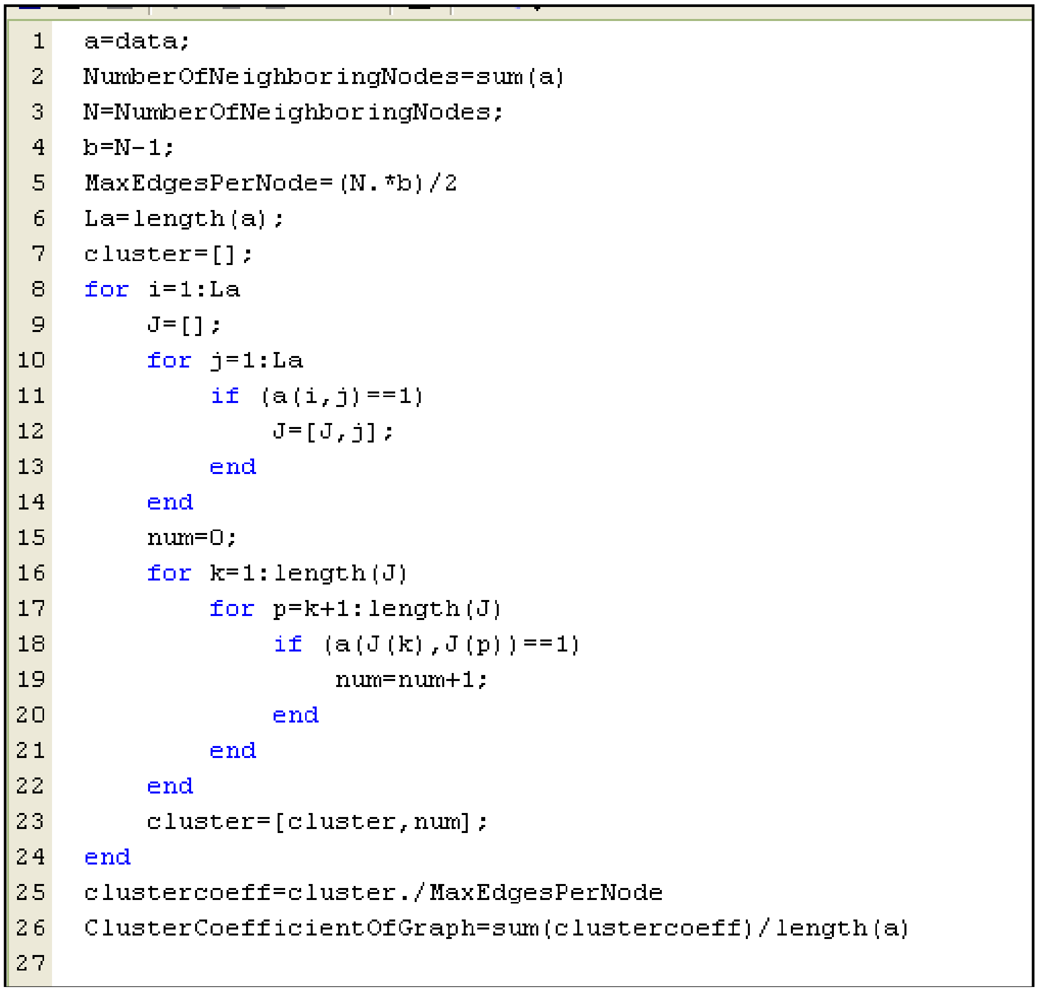
```
a=data;
La=length(a);
for p=1:La-1
for i=1:La
    for j=i+1:La
        if (a(i,j)==0)
            for k=1:La
                    if (k > i && k < j) %to search only at lower triangular matrix%
                           if(a(k,i)~=0 && a(j,k)~=0)
                                b=a(k,i)+a(j,k);
                                if b==p
                                    a(i,j)=b;
                                end
                           end
                    elseif (k < i && k <j)
                            if (a(i,k)~=0 && a(j,k)~=0)
                                   b=a(i,k)+ a(j,k);
                                   if b==p
                                      a(i,j)=b;
                                   end
                            end
                     end
            end
        end
    end
end
a=triu(a)+tril(a')
end
   a
```

Figure A – MATlab Algorithm for computing of Shortest Path Lengths of individual nodes



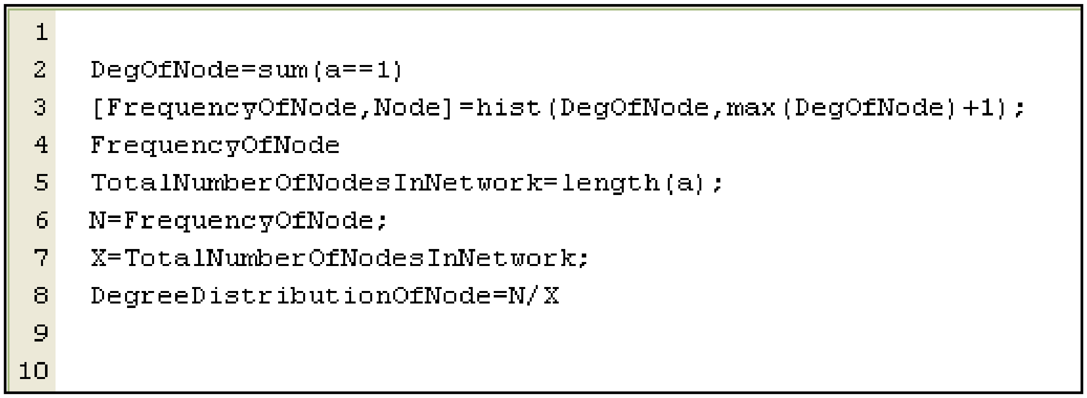
```
TotalNodesInNetwork=length(a)
Z=TotalNodesInNetwork;
MaximumNumberofEdgesPossible=(Z*(Z-1))/2
X=MaximumNumberofEdgesPossible;
b=sum(triu(a));
EdgesInNetwork=sum(b)
E=EdgesInNetwork;
AveragePathLengthOfNetwork=E/X
```

Figure B – MATlab APL Algorithm



```
a=data;
NumberOfNeighboringNodes=sum(a)
N=NumberOfNeighboringNodes;
b=N-1;
MaxEdgesPerNode=(N.*b)/2
La=length(a);
cluster=[];
for i=1:La
    J=[];
    for j=1:La
        if (a(i,j)==1)
            J=[J,j];
        end
    end
    num=0;
    for k=1:length(J)
        for p=k+1:length(J)
            if (a(J(k),J(p))==1)
                num=num+1;
            end
        end
    end
    cluster=[cluster,num];
end
clustercoeff=cluster./MaxEdgesPerNode
ClusterCoefficientOfGraph=sum(clustercoeff)/length(a)

```

Figure C – MATlab CC Algorithm



```

DegOfNode=sum(a==1)
[FrequencyOfNode,Node]=hist(DegOfNode,max(DegOfNode)+1);
FrequencyOfNode
TotalNumberOfNodesInNetwork=length(a);
N=FrequencyOfNode;
X=TotalNumberOfNodesInNetwork;
DegreeDistributionOfNode=N/X


```

Figure D – MATlab DGD Algorithm