
Abstract:

The most popular explanation of the emergence of scaling in the World Wide Web is the preferential attachment of the Barabasi-Albert network model. However, this explanation is challenged by the global information hypothesis, the existence of linear preference and the Matthew Effect in the World Wide Web. The project replicates a new directed network model, the Hidden Tree model, proposed by Bojin Zheng et al. in 2011 and proposes a few variants to analyse the accuracy and explore the potential of this model. The hidden tree model adds the hidden tree structure of the websites in the World Wide Web into the Erdos-Renyi model with a new rule: the source node needs to connect with all of the nodes on the shortest path from it to destination node. It combines the advantages of preferential attachment with a hierarchical structure. Based on the experimental results, the model is confirmed to achieve a network with power law distribution and small-world phenomenon.

Variants are added for the different theories of existing network models, such as the Price's model or "adding+walking" model. Variants are also designed for the challenges of Barabasi-Albert model, such as the appearance of the new big companies. Moreover, some variants are designed for a more complex network in order to test the stability of the hidden tree structure. These experimental results prove the success of this alternative explanation to the emergence of scaling in the World Wide Web and solve some challenges of the Barabasi-Albert model.

Acknowledgement:

I would like to thank:

- My Project Supervisor, Dr. Steve Gregory, for his invaluable advice, guidance and help.
- My parents and friends for their encouragement.
- My markers, Dr. Julian Gough and Dr. Ian Holyer, for their important feedback and advices.

Contents

Abstract.....	2
Acknowledge.....	3
1 – Introduction.....	6
1.1 – Aims and Objects.....	6
1.1.1 – Theory Analysis Objectives.....	6
1.1.2 – Software Experiments Objectives.....	6
1.2 – Dissertation Structure.....	7
2 – Theoretical Background.....	8
2.1 – World Wide Web.....	8
2.1.1 – the World Wide Web with power law distribution.....	8
2.1.2 – Growth Dynamics of the World Wide Web.....	9
2.2 – Poisson Random Graphs.....	10
2.2.1 – Erdos-Renyi Model.....	11
2.2.2 – Small-World Model.....	12
2.3 – Growth Model of network.....	13
2.3.1 – Simon’s Model.....	14
2.3.2 – Barabasi-Albert Model.....	15
2.4 – Hidden Tree Structure.....	19
3 – Experiment.....	26
3.1 – Building Hidden Tree.....	27
3.2 – the results of Hidden Tree model.....	28
3.3 – The variants of Hidden Tree model.....	32

3.3.1 – “the node with role”	32
3.3.2 – Random Connecting Probability.....	35
3.3.3 – Random number of branches.....	38
3.3.4 – Hidden Tree model with random branch number and role.....	41
3.3.5 – Hidden Tree with Role and Random Activity.....	43
3.3.6 – The tree model with random activity, structure and role.....	46
3.3.7 – Hidden Graph.....	47
3.4 – Wealth Model.....	51
3.4.1 – Matthew Effect.....	52
3.4.2 – Models and Experiments.....	53
3.4.3 – Wealth model analysis.....	60
3.5 – Hidden Tree model analysis.....	61
4 – Summary.....	62
5 – Future Work.....	63
Reference.....	65
Appendix.....	69

1. Introduction:

This section is to describe the aims and objects of the project. Then, a brief structure of the dissertation will be introduced.

1.1 Aims and Objects

There are two main objects in this project: the theory analysis objectives and model experiments objectives. The researching of the existing network model and the World Wide Web helps the design of hidden tree model which needs to consider different circumstances in the real world. Some models are just for a specific situation, so the theory analysis increases the content of software experiments. And with the result of experiments, analysing the variants of the hidden tree model in theory is to explore the potential of hidden tree. In the second object, the model experiments are to test, compare and develop the hidden tree model with other existing network models, in order to prove the hidden tree model can produce a power law distribution network and keep the features in more complex situations.

1.1.1 Theory Analysis Objectives

To study the following points:

- Since the Barabasi-Albert model is proposed, Erdos-Renyi model is not usually to be used to explore the World Wide Web. The power law distribution becomes an essential angle to explore the World Wide Web. As a complex network, the World Wide Web is not likely to be explained by Barabasi-Albert model itself. Thus, there are more models to explain the growth of the World Wide Web.
- The algorithm of Erdos-Renyi model cannot explain the emergence of scaling in the World Wide Web, but it still has the key feature: randomness. As the prototype of hidden tree model, why Erdos-Renyi model has a different result? Analyse the algorithm and structure of them to confirm the importance of the website structure in the World Wide Web.
- Barabasi-Albert model's challenges reflect in the unrealistic information collection and poor explanation for the fast arising websites. Thus, analyse the Barabasi-Albert model and the variants of it to solve these specific challenges to find the reason and explanation.
- The scaling of the World Wide Web produces many different circumstances. Use the software to simulate these circumstances to collect the results for following analysis.
- Collecting and analysing the results of variants of the Hidden Tree model to explore the potential of it.

1.1.2 Software Experiments Objective

The platform of software experiments is a JAVA language software name BlueJ which is

produced by the University of Kent. Using this platform, the program includes:

- Producing both an example of the hidden tree and an example of the traditional tree, then ensuring that the hidden tree still contains all of the basic features of the traditional tree.
- Importing the value of parameters to test the result of scaling in different conditions. For example, for a basic hidden tree model, the parameters are the number of nodes, the branches of tree and the probability of connection for each node.
- Producing variants designed for different situations in the real networks or that follow the rule of existing network models, in order to check the universality of the hidden tree model. The variants have considered the complex structure, different functions of web pages, direction in the network and the algorithm used in the directed graph model.
- Show the results as a scatter gram which clearly shows the distribution of nodes.

1.2 Dissertation Structure

In this report, the background including a brief study of the World Wide Web and the explanation of Erdos-Renyi and Barabasi-Albert model with the previous experiments will be described in section 2. At the same time, several important network models will be described in section 2.

In theory analysis objective, through the study of Erdos-Renyi and Barabasi-Albert model, the main features of the model and the variants to explain the emergence of scaling in the World Wide Web are discussed and the assumption to prove in software is evaluated. The details of theory analysis objectives and software experiments objectives of the Hidden Tree model are described in section 2 and 3.

In another object, all of the algorithms, codes and models will be explained. Moreover, the hidden-tree structure will be discussed with the other existing network models, Power law distribution, small-world and the other problems appearing in the Barabasi-Albert model through the algorithm and results of different model.

In section 4, the results of the experiments will be analysed with the result in the theory analysis section in order to discuss the hidden tree model for explanation of emergence of scaling in the World Wide Web on both technology and theory aspects. Furthermore, the evaluation of results can prove the success of the project.

Finally, future work is to discuss the development of the current hidden tree model in a more complex environment and other fields relating with networks and combine with different fields' theories to gain an integrated theory for special circumstance in the real world, such as the appearance of fast developing website.

2. Theoretical Background

2.1 World Wide Web

The World Wide Web (usually called World Wide Web or W3) published by Tim Berners-Lee in 1990[7] is a wide-area system with hypertext and hypermedia information aiming to build the connection and transformation in a universal area [8]. For the World Wide Web, some protocols and web languages are created, such as the Hyper Text Transport Protocol (HTTP) and Hyper Text Mark-up Language (HTML).

2.1.1 The World Wide Web with Power Law Distribution

Nowadays, Power Law Distribution is not only a phenomenon of the World Wide Web but also a key piece of evidence to prove the success of a network model. With the development of the World Wide Web, scientists found the result of traditional network models, random network model, conflicts with the reality, so some new theories were created. For example, in 1999, Barabasi, Albert and Jeong started to explain the scaling in the World Wide Web and created the Barabasi-Albert model. After that, more and more theories and models were proposed to address the challenge of the Barabasi-Albert model and improve the study of the World Wide Web.

The actions in web include random adding, growing and deleting, which is why people think web follows Poisson distribution. However, the unregulated growth leads the web into a huge directed graph, in which the vertices are the web-pages and the link between web-pages act as the edges. The World Wide Web is a continuously changing directed graph and the least number of vertices is 8×10^8 . Therefore, the research of the World Wide Web cannot collect the whole information including the vertices and edges. Most experiments use a search engine or ranking website to get the information. For example, Barabasi, Albert and Jeong proved the World Wide Web follows power law distribution by using a commercial search engine named Northern Light which just can cover 38% of the web which has been the largest coverage [4]; and Bernardo and Lada used Alexa and Infoseek crawls to collect the number of web pages [5], but the largest number of web pages is no more than 10^5 . This situation has not changed a lot recently, so the study of the World Wide Web is always hard to cover the entire network, just for the main part.

A significant experiment proving the power law distribution of the World Wide Web used local connectivity measurements to construct a topological model of the World Wide Web [4]. In this experiment, scientists used the model which maps the internet infrastructure to analyse the data distribution in searching for information. They collected all of the URLs which can be understood as the hyperlink or address of a web page. A common sense idea is a web-page contains the incoming and outgoing links, which is expressed as the directed edge on the web graph. Therefore, after collecting data, the probability of a web-page which has k in-degree or out-degree is named as $P_{in}(k)$ or $P_{out}(k)$. And the result of the probability is shown in Fig.1 which can clearly find the relationship between the probability and links'

number following a power law distribution. This result contradicts the theory of the Erdos-Renyi model with Poisson distribution.

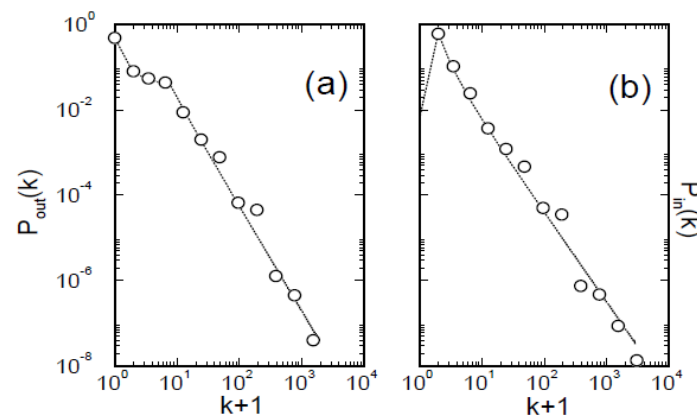


Fig.1 (a) the distribution of outgoing link (b) the distribution of incoming link, copied from [4]

Therefore, Power Law distribution became to be a key piece of evidence to prove the success of a network model

2.1.2 Growth dynamics of the World Wide Web

After this new discovery, more scientists started to find not only the incoming and outgoing links of pages but also the growth of web support that the World Wide Web follows power law distribution.

The World Wide Web is a dynamic exponential growing network constituted by websites and web pages. The growth of the web is the growth of websites and adding new website into the network. The increasing tendency of the World Wide Web is similar with other subjects, such as the society, technology etc. A famous example to explain the network is the collaboration graph of a movie actor. The actors can be separated into different years based on their first movie being shown. The connectivity of actors will increase for many years. In the network model, each actor is a vertex and two actors have a connection if they worked in the same film. The probability of an actor having k connections follows the power law distribution, $p(k) \sim k^{-\gamma_{actor}}$, where $\gamma_{actor} = 2.3 \pm 0.1$. This probability function can be used in other fields. For example, the electrical power grid of the western United States follows the power law distribution with $\gamma=4$ and the probability of a paper cited by k times also follows the power law with $\gamma=3$ [2]. For the World Wide Web, a network with 800 million vertices, the probability of a random vertex being pointed by other k vertices is following the $p(k) \sim k^{-\gamma}$, where $\gamma = 2.1 \pm 0.1$. On the other hand, the topological properties of networks generated by evolutionary dynamics are also studied by some authors [31][32] based on the Bak-Sneppen model[33]. The result of the model with a fixed or variable number of nodes showed the distribution of connection with the exponential or power law distributions. In the process of growth, the number of pages for each website also follows the power law distribution: many websites have few web pages while few websites have a large number of

pages. To explain this phenomenon, a stochastic dynamical growth model is needed [5].

From the data of Alexa and Infoseek, the result (Fig.2) expresses the power law distribution existing in the scaling in the World Wide Web.

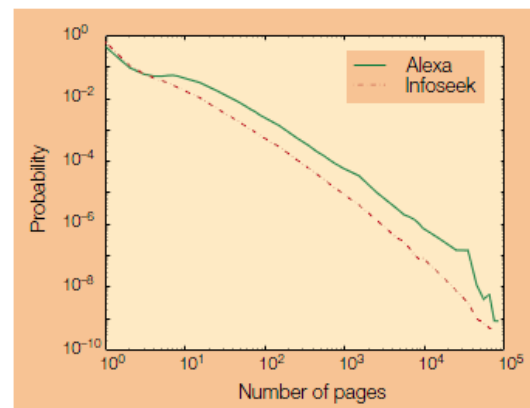


Fig. 2 the distribution of pages in each website, copied from [5]

In the process of scaling, the fluctuation of pages' number is related with the existing number of websites, which means a site with a million pages has high probability to gain or lose hundreds of pages but a small website with only tens of pages is not likely to get a big change in the amount of pages. And the fluctuation of pages can be assumed as a random fraction of the amount of pages. This phenomenon is explored in the future network model with the effect of website's age, such as BA model. If the growth rate of websites is similar or equal but the individual fluctuation is a random value, the amounts of pages in each website are still different and lead to a log-normally distribution after a long period of time.[6] To explain this, the random variable k is the number for the adding or deleting of a website and n is the amount pages in a given day, so after m days, the amount may be $n_{(now)} = n \times k_m$. However, this log-normally distribution is just similar with the power law distribution. The analysis of the growth of the web still needs to consider another two factors: the age of website and growth rate.

There are more young sites than old in the World Wide Web. And it is not hard to find an old website having a large number of pages while it is hard to find two websites with the same growth rate. Thus, if the effect of age and growth rate is added into the growth of web, the probability to find a website with a given number follows the power law distribution like Fig.2.

2.2 Poisson Random Graphs

A decade before Erdos and Renyi studied random network integrally and systematically and named it as "Random Graphs" by which it is most often known today, Rapoport and collaborators had started to study how to construct a large and random network called "Random net"[39][40]. The Random Graphs can also be called "Bernoulli graph" [41]. In this

section, it is called Poisson Random Graphs is to distinguish between models with different degree distribution.

Like the Erdos-Renyi model, most of Poisson Random Graphs is inadequate to explain some circumstances of networks in real-world, such as the power law distribution in the World Wide Web, but some extension of model can lead to other degree distribution. The Hidden Tree model, the main research objective in this project, is one of the extended models of ER model.

2.2.1 Erdos-Renyi model

Erdos-Renyi (ER) model is a random directed graph model named for Paul Erdos and Alfred Renyi[9]. The graph produced by ER model follows the Poisson distribution, so the graph of ER model can be called as Poisson random graph.

The model is creating number n of vertices as the initial situation and connecting these vertices in a probability p . In this model, there are two kinds of expression, one can be expressed as $G_{n,p}$ and the other one is $G_{n,m}$. These two functions are defined by three parameters: the number of vertices, the number of edges and the probability of the graph appearing. $G_{n,p}$ means all of the graphs having m edges. The probability of the graph is $p^m(1-p)^{M-m}$ while $M = \frac{1}{2}n(n-1)$ the probability is the maximum. $G_{n,m}$ is to get the graph with n vertices and m edges. In $G_{n,m}$, all of the graphs have equal probability to appear.

As a limit large scale graph, the ER model can explain many features of the network [9][10][11]. The network does not just mean the World Wide Web. In the network topology, the vertices are the elements of the system, and the edges are the interactions between them. For example, in different networks, the vertices and edges present different objects. In gene network, vertices are proteins and genes, the edges are the chemical interaction; in nervous systems, the vertices are nerve cells and the edges are the axons; in social, the vertices are individuals and organizations and the edges are social interactions; in the World Wide Web, the vertices are HTML documents and edges are links. The mention of these networks is to explain that the World Wide Web is not a simple random network. It has some special features such that it follows the power distribution, so scientists create new theories or design the variants of the ER model. This is a reason why the Hidden Tree model is still developed from a model which has been proved unable to explain the World Wide Web. In ER model, each vertex has individual probability to get connection, so the probability of a vertex having degree k is $P_k = \binom{n}{k}p^k(1-p)^{n-k} \cong \frac{z^k e^{-z}}{k!}$ in which p is the probability, k is the degree and z is the mean degree equalling $p(n-1)$. The result of the degree distribution is clearly Poisson distribution.

In the real network, there are different ways to connect vertices, such as in the World Wide Web, the pages are connected through the path. Thus, the probability of each vertex being able to be connected is different. In the model of Erdos and Renyi, they used a phase

transition to process the graph. Phase transition is to develop the graph from a low-density and low- p into high density and high- p state in which the extensive state is the whole vertices connected with each other to generate a single giant component, and the rest vertices groups into small components with an exponential size distribution and finite mean size [12].

The scale of the giant component can be calculated by a simple heuristic argument. This argument can also attempt to calculate the expected diameter of the World Wide Web. Because it is the earliest network model, the following network models are influenced by its opinion. The probability of vertex selected from the graph out of the giant component, u , is equal to the fraction of vertices which is not in the giant component. If the vertex has k degree, the probability of it not belong to the giant component is u^k which is also the probability of all of its neighbour vertices which do not belong to the component [12]. With the expression of probability of a vertex having degree k , the u in a finite scale large graph can be calculated by using: $u = \sum_{k=0}^{\infty} p_k u^k = e^{-z} \sum_{k=0}^{\infty} \frac{(zu)^k}{k!} = e^{z(u-1)}$, so the fraction of component in the graph is $S = 1-u$. Therefore, the mean scale of giant component is $\langle s \rangle = \frac{1}{1-z+zS}$. It is easy to find the solution of S . Because z must be no less than 0, the solution of the expression is the expected scale of giant component.

Even though most of the random graphs do not match the real world networks, there are still some variants which can explain the features of networks, such as the small-world model.

2.2.2 Small-World Model

The small-world model proposed by Watts and Strogatz[24][42][43] is a simple and effective network model. A typical demonstration of small-world effect is that if letters passed person to person, the number of steps from sender to recipients is a small number around six[44].

In mathematics, the mean distance between a pair of vertices is $l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij}$

where d_{ij} is the distance between vertex i and j . Thus, the result proves the vertices prefer to select a short path to connect with each other in most of networks.

Scientists assume the network has geographical features, so the vertex in the network has a position. The distances between vertices may decide which vertex is connected by another vertex. In small-world model, the network is built as a low-dimensional lattice and the edges are added or moved in order to create a low density path which is added into another lattice. The small-world model can build a network in any dimensional. To research the effect of distance clearly, the low-dimensional lattice is a better choice, so most of experiments use the one-dimensional lattice. In a one-dimensional lattice, if there has L vertices and each vertex connects with its neighbours k or fewer lattice spacing away, the graph has L vertices and Lk edges. Then, the small-world model is created by taking a small fraction of edges and reconnecting them. The process of reconnecting is to relocate the ends of each edge in a p probability to another space in the lattice randomly. In this process, there has no two edges

connecting same pair of vertices or the vertex connecting to itself. If the probability is 0, the graph does not change, so it cannot show the properties of small-world. It is a regular lattice with clustering coefficient $C = (3k - 3)/(4k - 2)$ which tends to 3/4 for extreme k . If probability is 1, all of the edges are reconnected to new locations, which lead a random network. The mean distance is $\log L / \log k$ and the clustering coefficient is $C \cong 2k/L$ [12].

To analyse small-world model of Watts and Strogatz in mathematics, the model's rules are simplified. There can has two edges connecting the same pairs and the vertex can link to itself. In this situation, the result graph is between regular lattice and random graph. After Watts and Strogatz, the variants models proposed by Monasson[45] and by Newman and Watts[46] are very popular. Both variants replace the reconnecting process with adding the pair of vertices chosen randomly into the low-density lattice, so the probability of the edge which is selected controls the density of lattices. Therefore, the mean total number of edges is Lkp , the mean degree is $2Lk(1 + p)$ and the mean distance is in a limit range.

The degree distribution of small-world model does not suit for the World Wide Web which was proven by Barabasi and Albert. For the original model, each vertex has k edges, so the degree is k . Thus, the probability p_j of having degree j is [47]:

$$p_j = \sum_{n=0}^{\min(j-k, k)} \binom{k}{n} (1-p)^n p^{k-n} \frac{(pk)^{j-k-n}}{(j-k-n)!} e^{-pk}, \text{ for } j \geq k$$

$$p_j = 0, \text{ for } j < k.$$

For the variants, the expression is simpler, that is [47]:

$$p_j = \binom{L}{j-2k} \left[\frac{2kp}{L} \right]^{j-2k} \left[1 - \frac{2kp}{L} \right]^{L-j+2k}, \text{ for } k \geq 2k$$

$$p_j = 0, \text{ for } j < 2k.$$

2.3 Growth model of network

In this section, the network models attempt to explain the properties of network. These models produce the networks which grow smoothly by adding new vertices and edges, in order to map the network in real world. At the same time, simulating the process of growth is also help to discover the structural features of network. For example, scientists used the "triadic closure" process to design the model to study the transitivity of network [38] [48][49]. In these models, edges are prefer to linked with a pair of vertices having a common neighbour in order to complete a triangle, which may increase the amount of transitivity.

However, the most popular research aim is to explain the power law distribution of network. A basic network model includes the number of edges which is represented as k and the

probability of distribution is $P(k)$. A brief definition of power law theory is $P(k) \sim k^{-\gamma}$. In this section, the Simon's model which explains the power law distribution with "richer get richer" will be described, and the highly influence model of Barabasi and Albert is discussed.

2.3.1 Simon's Model

The first model explaining the power law scaling was proposed by Herbert Simon in 1955[17]. Simon's model is designed for general scaling phenomenon rather than network system, but it can also explain the power law distribution of the World Wide Web after combining with the features of internet growth. In some ways, the most popular power law distribution model, Barabasi-Albert model, is the special case of the Simon's model.

In the Simon's model, the factors affecting scaling are distributions of frequencies of the words appearing in texts or population figures of cities [28]. Thus, in the World Wide Web, a web page can be seen as a word and the frequencies are the number of hyperlinks linking to it. The growth of a system is the process to increase the counters which can be seen as the words and their frequencies or the nodes and their activities. Thus, the growth of a system is to add new elements and enhance the frequencies of existing elements in a rate proportional to their current state.

In this model, there are n nodes and each node is linked by k_i links from the class $[k]$: k_i and $i \in [1, 2, \dots, n]$. The process of a network growth is iterating the steps [17] below:

- (i) A new vertex has a probability α to be linked to an existing node.
- (ii) Using the function below to select vertices from class $[k]$ and connect with a random vertex.

$$P_{\text{new link to class } [k]} \propto kf(k). \quad (1)$$

Because it is a stochastic process, it is not hard to find a solution which can satisfy all of the situations.

$$\gamma = 1 + \frac{1}{1-\alpha} \quad (2)$$

α is the relative growth of the number of vertices and edges, so it should be a small number. And the γ could be treated as 2.[17]

To explain the emergence of scaling in the World Wide Web, the data from Altavista crawls was collected. Thus, the model can get the predict exponent of the link distribution γ is 2.1 in which α is about 0.10. This result is close to the Barabasi and Albert's experimental result $\gamma = 2.1 \pm 0.1$ [2].

2.3.2 Barabasi-Albert Model

The World Wide Web contains implicit mechanisms rather than a pure random network that is why ER model leads to a different distribution. In 1965, Derek de Solla Price[41] described probably the first example of scale-free network. He proposed a mechanism of cumulative advantage to explain the power law distribution of networks. However, Price's model and the cumulative advantage theory are not large known by people. In 1999, Barabasi and Albert rediscovered the theory. They used the data of commercial searching engine to map the World Wide Web model and found the Power Law distribution, which is against the result of ER model. After that, they named the cumulative advantage as "preferential attachment" and Barabasi-Albert(BA) model becomes the most extensively accepted theory in hundreds of complex network models explaining the power law distribution or growth of the World Wide Web. This discovery breaks the common sense that the World Wide Web is a Poisson distribution graph lasting for 50 years.

BA model is similar to Yule Process[14, 15], Price's model[16], and a particular case of Simon's Model[17]. The model produces an undirected network graph, so there is no different between in-degree and out-degree. The advantage of the undirected network graph is the BA model does not the worry about the problem of how a web-page gets its first link [41]. The disadvantage is the real World Wide Web is a directed graph, so the undirected graph model has to miss some features which may affect the accuracy of result.

At the beginning, the initial graph contains few vertices and edges. In the model for the World Wide Web, the vertices are the web-pages and edges representing the connection between web-pages. With the growth of the network, the vertices continuously connect with each other and new vertices add into the model. Connecting new nodes and existing nodes depend on the degree of existing nodes, which means an old node having more edges has a high probability to be connected. This phenomenon is the preferential attachment. In fact, this theory is a variety of the Matthew Effect [13] which is a positive-feedback phenomenon [3].

To explain the preferential attachment in mathematics, if p_k is the fraction of vertices in k degree, the probability that a vertex with k degree is connected with another vertex is:

$$\frac{kp_k}{\sum_k kp_k} = \frac{kp_k}{2m}$$

In this equation, $2m$ is the mean degree of the network, because the network of BA model is an undirected graph, if there are m edges added into the vertex, the both ends of edge are used to increase the degree. Thus, if the network adds a new vertex with m edges, the mean number of vertices which is linked is $m \times \frac{kp_k}{2m} = \frac{1}{2}kp_k$. At the same time, the distribution of degree is changed, because the selected vertices' degree is increased to $k+1$ and the number of vertices is added. Therefore, when network has n vertices, the probability is:

$$(n+1)p_{k,n+1} - np_{k,n} = \frac{1}{2}(k-1)p_{k-1,n} - \frac{1}{2}kp_{k,n}$$

If $k=m$, $\frac{1}{2}kp_{k,n}$ is equal to 1. Thus, we can get the equations of p_k is:

$$p_k = \begin{cases} \frac{1}{2}(k-1)p_{k-1} - \frac{1}{2}kp_k, & \text{for } k > m \\ 1 - \frac{1}{2}kp_k, & \text{for } k = m \end{cases}$$

So, $p_k = \frac{(k-1)(k-2)\dots m}{(k+2)(k+1)\dots(m+3)} p_m = \frac{2m(m+1)}{(k+2)(k+1)k}$. In the finite network, it can get $p_k \sim k^{-3}$, so it means that the degree follows the power law distribution with $\alpha = 3$. However, in this model, the value of α is fixed[29].

The network of the BA model has two important features [19]. The first is that the relationship between age and degree. It is easy to know that the degree of the vertex increases with the growth of network. Thus, the earliest vertices must have higher mean degree than new vertices. However, this result is used to argue against the BA model by Adamic and Huberman[20], because there is no correlation in the World Wide Web.

In the experiment of Adamic and Huberman, they believed that each site has individual growth rate. They collected the data of 260,000 sites and the distribution of links the sites received from other sites followed the power law distribution (Fig.3). Then, they found there is no correlation between age and degree, after they got the registration data of each sites. In their research, a popular site registered in 1999 has more links than an old site created in 1993 [20].

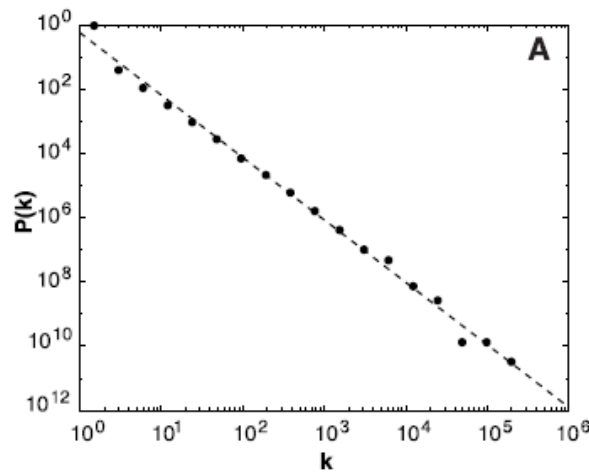


Fig.3 The distribution function for the number of links. Copied from [20]

The second feature is that the degree of the vertex is affected by the adjacent vertices. Krapivsky and Redner found that in a pair of vertices, if one vertex had in-degree j and out-degree k , the probability distribution is:

$$e_{jk} = \frac{4j}{(k+1)(k+2)(j+k+2)(j+k+3)(j+k+4)} + \frac{12j}{(k+1)(j+k+1)(j+k+2)(j+k+3)(j+k+4)}$$

However, in the BA model, it is an undirected network, so $j+k$ is the total degree of vertices.

Since proposing the BA model, the challenges are never stopped. In this project, there are four challenges are concerned.

First, each new node must know all of the information of the whole network, which is called the global information hypothesis [21]. Obviously, it is impossible for a large scale network, because of the huge quantities of data, let alone the complex network such as the World Wide Web which contains millions of web pages and the billions of information to be collected, stored and managed.

To solve this challenge, many scientists proposed the variants to correct the BA model. For example, Alexei Vazquez proposed “adding+walking” to get over the problem [21]. Alexei Vazquez thought that a new node did not have the whole information of the network. It needed to collect information through adjacent vertices. For instance, the new node is like a manuscript in a paper. The author does not know all of the papers including the manuscript, so he has to get more information through the reference on the papers which he knows.

In the model, the initial condition is only one node $N = 1$ and an empty set of links. The loop steps are adding nodes and walking through the network. “adding” is to add a new vertex into the network and link to an existing node randomly. “walking” is the new vertex creates links with a probability p to the nodes which link to the node selected in the “adding” step.

In Fig.4, it is a run of algorithm with $p=0.5$ and the maximum nodes number is $N=5$. At the first, a node is created called node 1 and $N=1$. Then, the second new node (node 2) points to node 1 and the N increases to 2. When $N=3$, the node 3 can select the node to point. In this figure, node 3 points to node 1. Then, node 4 can also select node from candidate nodes 1, 2 and 3. In this case, it selects node 2. Now, node 2 has a link to node 1, so node 4 links to node 1 with the probability is p . Node 5 is created and points to node 4 which has a link to node 2. Like the previous step, node 5 creates a link to node 2 with probability p . Because node 2 has a link to node 1, node 5 also need to link to node 1 with probability p .

Unlike the BA model, Alexi Vazquez’s model produces a directed graph[38], which means the node has in-degree and out-degree. To calculate the degree distribution, the in-degree is defined as the number of links pointing to the node. Another difference with the BA model is

the in-degree of new node is 0. If there is a new node, the in-degree k of node increases 1. Thus, if the probability is $w(k, N)$ when adding node $N+1$, the number of nodes with links can be computed by using equations below:

$$n(0, N + 1) = n(0, N) + 1 - w(0, N)n(0, N), (1)$$

$$n(k, N + 1) = n(k, N) + w(k - 1, N)n(k - 1, N) - w(k, N)n(k, N), \text{ for } k > 0. (2)$$

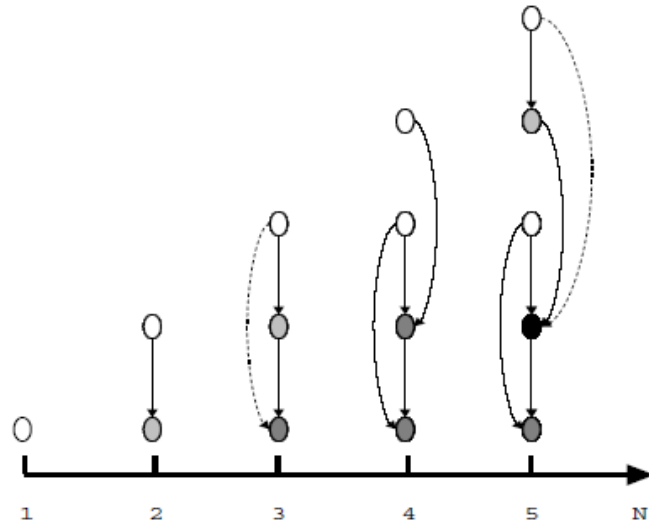


Fig. 4 The generation of network with 5 nodes running up. Copied from [21]

If $N \gg 1$, the equations can be solved as $w(k, N) = W(k)/N$, where $1/N$ is the probability of a new node adding into the network. So, it can also get $n(k, N) = NP(k, N)$. Because of the stationary condition $P(k, N + 1) = P(k, N) = P(k)$, the value of $P(k)$ is:

$$\begin{cases} P(0) = \frac{W(0)}{2}, \text{ for } k = 0 \\ P(k) = \frac{W(k-1)}{[1 + W(k)]}, \text{ for } k > 0 \end{cases}$$

Thus, it is not hard to get the results:

$$\begin{cases} P(k) = 2^{-(k+1)}, \text{ for } p = 0 \\ P(k) = [(k+1)(k+2)]^{-1}, \text{ for } p = 1 \end{cases}$$

When $p=1$, $w(k, N) = (1 + k)/N$ which leads a preferential attachment phenomenon. However, in this model, it is just a special case.

The second challenge has been discussed before. The rational action of connecting new node needs to consider not only the preference but also the linear rule, but this action leads

to the correlation between age and degree [20,22,23].

Besides the experiment of Adamic and Huberman, Bianconi and Barabasi also gave an extension of the BA model to explain there is no relationship between age and degree [25][26]. In their model, each new vertex is given a parameter η_i named “fitness” which affects the ability to attract links. Thus, the connection depends both the degree of the vertex and distribution $p(\eta)$. However, the value of fitness is limit, which means the distribution $p(\eta)$ should be finite. Otherwise, the vertex with high η_i may be linked by most of vertices-a sort of “winner takes all” phenomenon [18].

The third challenge is that the BA model does not support the bounded distribution of Watts and Strogatz’s “small-world” [24]. Ravasz and Barabasi: hierarchical organisation model which tries to combine the property of modularity, high clustering and scale-free together.[27] In the experiment of Lada et al.[23], there was no correlation between the age and number of links, which is a different prediction to the BA model. A possible assumption is that the rate of acquisition of new links is associated with the links number a web site already has.

Lastly, the preferential attachment, richer-get-richer theory, enhances the importance of existing nodes especially for the high degree nodes. Thus, the small websites or companies will never beat the competition. However, the companies like Google or Facebook are very typical examples that grow quickly from the small companies to the large companies.

Though these challenges have appropriate solutions, none of these models can explain all of these challenges. In next section, the Hidden-Tree model is a new theory attempting to explain all of these challenges.

2.4 Hidden-Tree Structure

Preferential attachment has been introduced in front. In fact, the explanation of degree distribution of pages by using the Matthew effect is “the higher vertex degree because of higher degree of vertex”. This explanation sounds like a strange tongue twister, which is the reason that people think it is in a vicious circle. To change this idea, scientists need to find the more basic element to compose the complex network or the World Wide Web with power law distribution.

Bojing and his colleagues found a new theory to explain the emergence of scaling in the World Wide Web. They used the hidden structure of websites to study this emergence. There is an obvious fact: most of the websites are built as tree structure. The main page is the root and the specific pages are the leaves on the tree. In this structure, each web-page is a node in the model. When a node (leaf) connects to another node, the path may include the superior nodes of both connecting nodes, which means the source node points to all of the nodes in the path from itself to the item node. Thus, the information of these nodes is collected, which can lead to more useful data. This easily understood structure not only

offers convenience to human visitors but also lets the website maintenance become easier. The links in the tree are different with the links of the World Wide Web but they have the same data format. Moreover, this structure is hard to be picked up from the links. It looks like an invisible tree behind the website. Thus, this structure is called as hidden tree following to hidden order [37].

Therefore, in this paper, the Hidden Tree model which is the variant of ER model tries to explain the scaling of the World Wide Web. It assumes in the website all nodes (pages) are grouped into the hidden tree. Thus, there is a path including some nodes (pages) in the hidden tree between two pages. The path should be the shortest and the start node (page) should link with all of the nodes in the path. Therefore, the model's initial features can be defined as follows:

1. All nodes organised as a hidden tree.
2. Every node has the probability to connect with the random node.
3. The source node connects with all of the nodes in the path from source to destination node.

Same as ER model," every node would connect to the others with an invariant probability." [3] However, because all of the nodes are in the hidden tree, the source nodes not only link to the destination nodes but also link with all of the nodes on the shortest path. In this model, all of the hidden trees should be able to be connected to a big tree with a virtual root node. Thus, two aspects for this model are 1) algorithm for generating the tree, 2) algorithm for generating the network.

The hidden tree is assumed as an n-tree, in which n is the average number of child nodes for every node. When $n=2$, the hidden tree is a 2-tree. The number of whole nodes is named N. Each node has a variable quantity to show the times of being selected. After creating a stable structure of hidden tree, each node's level(rank) which is the distance between nodes is confirmed. For example, a node's level is 1 which can be shown as Rank(1), so the level of child node is 2 and 3 for grandchild node. Following this method, the hidden tree is constructed. The algorithm of building a hidden tree and the connection algorithm for each node is shown below [3].

Algorithm 1:

0. Initialize the variables tree and nodealt, dealt;

1. while nodealt is not empty

2. tbNum = k;

3. while tbNum > 0

4. rnd= random(0,1);

5. if rnd < tbNum

6. if nodealt is empty break; end

%Set the ranking number.

7. tree(dealt, nodealt) = Rank(Dealt(1)) + 1;

8. tmp = popup(nodealt);

%Attach the last element of nodealt to dealt.

```

9. attach(dealt,tmp);
a. end if
b. tbNum = tbNum -1;
c. end while
d. popup(dealt);
e. end while

```

Algorithm 2:

0. Initialize AdjMatrix and generate the hidden tree;

```

1. for i=1:N
2. act = activity;
3. while act > 0
4. rnd= random(0,1);
5. if rnd < act
6. nodenum = SelectANodeRandomly();
7. path = GetPath(i, nodenum);
8. AdjMatrix(i,nodenum)=1;
9. AdjMatrix(i,path)=1;
a. end if
b. act = act -1;
c. end while
d. end for

```

To simplify the experiment, there are four assumptions as the initial condition for the model. The first one is that the hidden tree is n-tree which means that each node has the same number of children web pages. The second is that each node has the same probability. Third, the hidden tree is static. Fourth, link all nodes in the path. However, in the real World Wide Web, the situation is much more complicated. Some websites have an irregular structure. For example, they may use a structure of a graph to replace the hidden tree or the tree is not an n-tree. Moreover, the probabilities of nodes may be different; the structure of the tree may keep changing with the growth of the web or the source node cannot link with all of the nodes in the path.

In the experiment of Bojin,Z. et al., the experiment model is very simple with only three parameters. The purpose of the experiment is to prove the degree distribution follows the power law distribution, so it is able to successfully explain the web scaling. As with the BA model, the degree distribution is $P(k) \sim k^{-\gamma}$ in which k is the degree of node and γ is a constant.

During the experiment, it is easy to find that these three parameters are the main points to generate a network obeying the power law distribution. Moreover, this model supports the theory of “small world”.

To validate the effect of the number of nodes N , we choose that $n = 2.0$, activity = 0.4 and $N = 1000, 2000, 5000, 10000, 20000$.

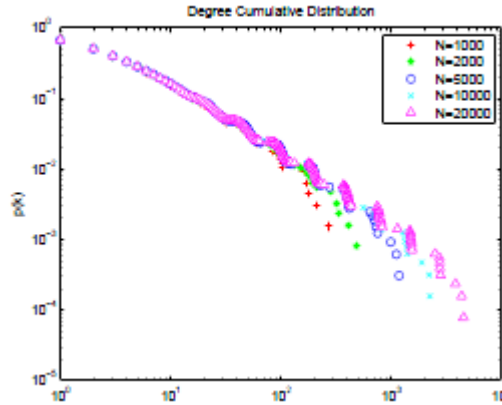


Fig. 5 Degree Cumulative Distribution When N Varies. Copied from [3]

In this figure, the exponent γ does not change with increasing the number of nodes N . And there is a cut-off at the end of the curve. The location of cut-off depends on the N . When N increases, the cut-off moves right, because of the limit of nodes. And the jump appears with the increasing of nodes number N . The jumps in this figure are because of the discrete level. Randomness leads the curve to drop down quickly, which means in the same degree, the degree of nodes may different.

To validate the effect of the number of child nodes n , $N = 10000$, activity = 0.4 and $n = 1.5, 2.0, 2.5, 5.5, 7.5$ are chose.[3]

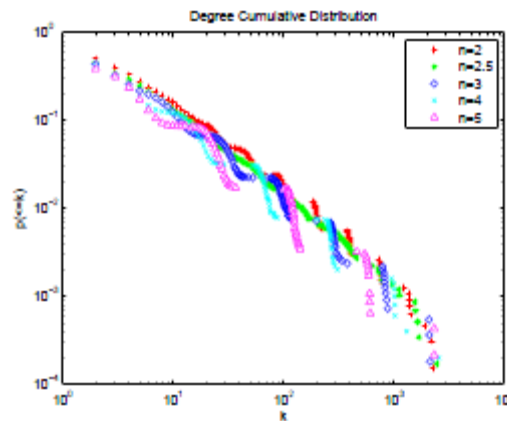
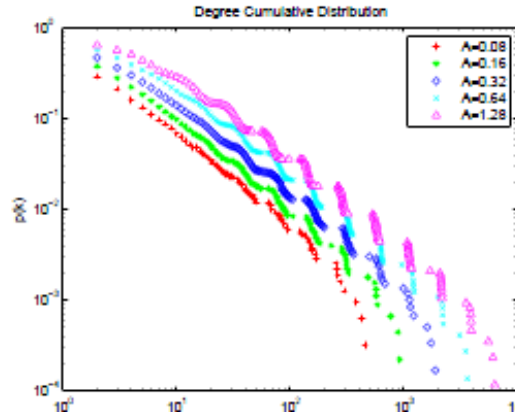


Fig. 6 Degree Cumulative Distribution When n Varies. Copied from [3]

In this figure, the curve is linear, so the in-degree part still follows the power law distribution for different n . With the increase of n , the phenomenon of interrupt is more obvious.

To validate the effect of the parameter activity or probability, $n = 2.0, N = 10000$ and activity $A = 0.08, 0.16, 0.32, 0.64, 1.28$. [3]



Dig.7 Degree Cumulative Distribution When activity Varies. Copied from [3]

Thus, based on the result above, the conclusion is that the in-degree distribution follows the power law distribution. When A is small, the curve is close to the ideal power law distribution curve.

Therefore, for different parameters n, N and A , the in-degree distribution always follows the power law distribution.

To prove that this model also satisfies the small-world effect, the decision parameters of small-world effect are clustering coefficient and shortest path. In the experiment, the relationship between clustering coefficient and activity (probability) will be tested. In theory, the value of activity can lead the style of the network to three different styles: community structure, small world structure and super small world structure. To express parameter activities relating to the small-world clearly, the N does not need a large value which will not affect the result. In the real world, the N is very large, but in this experiment, the large number of nodes may lead the graph to become confusing. Therefore, scientists choose $N = 300$, $n = 2$, activity = 0.04, and $N = 100$, $n = 2$, activity = 0.4 and 2.0. The results are in the following figures. Through the figures, it is easy to find that the larger clustering coefficient appears with smaller average shortest path, which is the evidence to prove the small world theory is satisfied in this model.

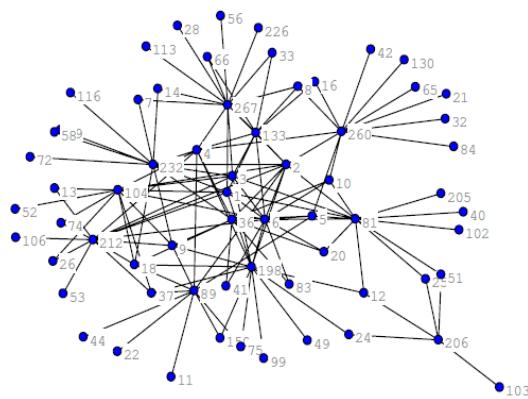


Fig.8 activity=0.04. Copied from [3]

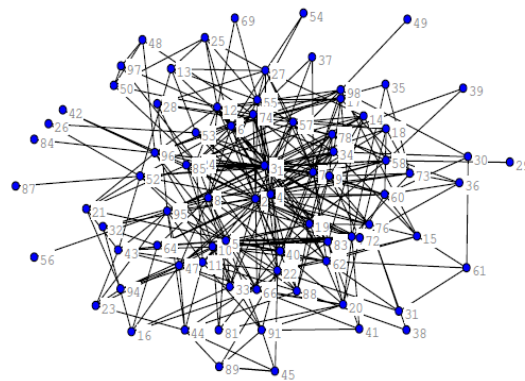


Fig.9 activity = 0.4. Copied from [3]

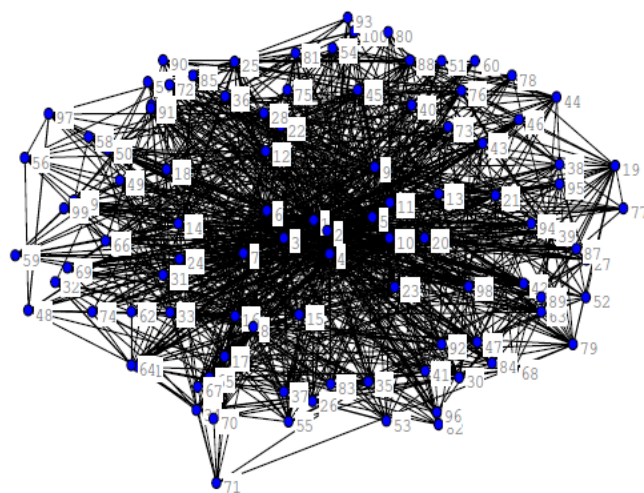


Fig. 10 activity = 2.0. Copied from [3]

In general, as the explanation of scaling in the World Wide Web, preferential attachment is accepted by most people. However, this scheme is not able to expel some challenge by itself.

This model uses the hidden tree and provides the following valuable results. The first is that this model generates a network obeying power law distribution which makes the explanation of the scheme for World Wide Web possible. Secondly, there is no global information hypothesis, so a node does not need to visit the information of all nodes. Thirdly, it has no obvious preference, but the hidden tree can be a kind of preference because the features of a tree make the high level nodes less than the low levels. On the other hand, it describes the wish of designers: the visitors may become interested in the common topics of the website. Fourth, small-world effect is controlled by only one parameter. Fifth, the phenomenon of Google or Facebook becoming popular can be explained. Because this model is not affected by time, the new companies can be added into the tree as a child tree which is called “niche”. Sixth, there is no time affection and the scale free network is stable because each node has the same action probability.

In this model, there still has the phenomenon with preferential attachment, although it does not focus on the preference. The higher connectivity nodes are more likely becoming the centre of the network. This phenomenon can be seen as the expression of basic fact of the Matthew Effect.

The main problem of this model is the jump at the end of the curve. This means the data of parameters should be adjusted or there are some other unknown schemes to fix this problem. Thus, it leads to another important idea that the nodes of the hidden tree should have a different status. In the real world, the nodes play different roles. The content in the main page of the website is a combination of links. Visitors access other pages through the main page and the details of the topic are in the child or grandchild pages. Therefore, different pages have different functions in the website, such as the index or content and the activities are different.

To test the affection of the node’s function for the distribution, the tree structure has to satisfy an inflexible rule: the content pages are linked to their father nodes; there is no content in page corresponding to the father node but the combination of links; the content nodes have no child nodes and father nodes can be control by its father. Therefore, the website is a tree, the content node is the leaf and the father nodes provide the link or path to these leaves.

However, in the real world, the structure is the variants of the hidden tree being built like the rules above. In this experiment, the fixed rule makes the result clearer. The network grows in this new rule, which means each leaf node has a probability to choose the destination nodes. Source node must link with all of the nodes on the shortest path. The other nodes do not do anything. The algorithm is[3]:

```
0. Initialize AdjMatrix and generate the hidden tree;  
1. LeafNodes = findLeaves(tree);  
2. for i=1:size(LeafNodes)  
3. act = activity;  
4. while act > 0  
5. rnd= random(0,1);
```

```

6. if  $rnd < act$ 
7.  $nodenum = SelectALeafNodeRandomly(LeafNodes);$ 
8.  $path = GetPath(i, nodenum);$ 
9.  $AdjMatrix(i, nodenum) = 1;$ 
  a.  $AdjMatrix(i, path) = 1;$ 
  b. end if
  c.  $act = act - 1;$ 
  d. end while
e. end for

```

The result of the experiment is in figure 11. In this figure, the power distribution is similar with the previous experiment's, even though the rule is changed and the roles of nodes are different. Thus, it can be confirmed that the functions of nodes does not affect the distribution.

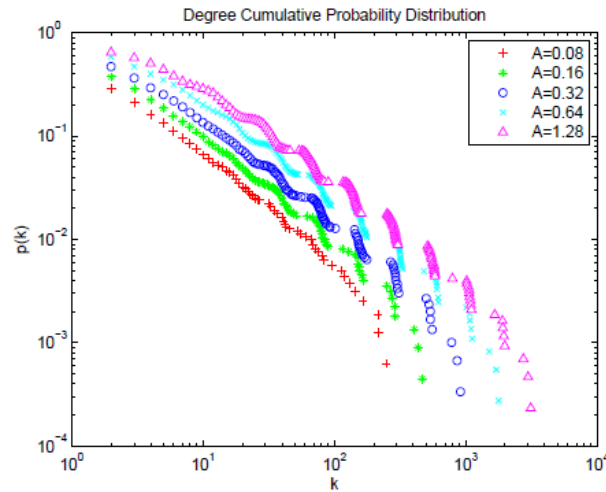


Fig.11 Degree Cumulative Probability Distribution when A is changing. Copied from [3]

3. Experiment:

The experiment of the Hidden Tree model follows the rules of traditional network model experiments. In the experiment, a hidden tree model acts as a website and the initial state is the number of vertices (web pages) and edges (hyperlinks). The only one difference is the vertices and edges in the hidden tree model constitute a tree structure rather than a random directed graph. The algorithm of the model is a little different to the hidden tree algorithm of Bojin et al in 2011.

The experiment is divided into three parts: the first part is to simulate the model of hidden tree; the second is testing whether the basic hidden tree network model can lead to a power law distribution or not; and the last part is designing the variants of the hidden tree model to explore the potential of the model.

3.1 Building Hidden Tree

In the original design, hidden tree model is a simple tree model which is implicit in the website. In the design of the website, the tree structure is easily found. For example, the index page is the root of the tree, the nodes of first degree are the index of different sections of website and the leaves are the pages with specific content. However, in many papers or books about websites, there is no definition about the structure. Thus, before building the hidden tree, it needs to be compared with a regular tree model, in order to check whether a regular tree contains the features of hidden tree or not.

In the program, each node of traditional tree contains all the information, such as the degree, parent nodes id, children nodes id and the id of itself. In hidden tree, the tree is a matrix; the value in the matrix is the relationship between nodes. During the developing of the programme, the traditional tree program is more flexible. The user can build and modify the tree at any time, while the hidden tree is a static tree, the node cannot be added or deleted. At the same time, the advantage of the hidden tree matrix is simple, which can enhance the efficiency. In the World Wide Web, a common website has a large number of pages, so the simple model is good for research. This result seems very silly in the project, because the hidden tree's matrix is built following the features of a tree. However, in the future work, the website must grow which includes the connection between nodes and the change in the number of nodes. Therefore, how to design a better algorithm to contain the features of growth should be a point in future.

3.2 the results of Hidden Tree model

Based on the result of building the hidden tree model, an essential aim of this project is to test the success of this model. This part did not collect information from different websites, because the aim is to test the hidden tree model to explain the distribution of degree in different conditions. This section also explains the reason why the Hidden Tree model does not have the problem of global information hypothesis and compare with Alexei's "adding+walking" model.

In the real World Wide Web, a successful website should let users surf the internet without using the backward or forward button on the browsers. Therefore, in this model, each original edge is a two-way path, or it can be seen as two links of two nodes pointing to each other. It does not mean it is an undirected tree. The more problems of direction will be researched in its variants. Like most directed network models, each node has in-degree and out-degree value. The initial value is in-degree=1 because each web-page has an address, if the degree is 0, it means nobody can access this page. This situation is also be faced by Price[30]. In Price's model, if the vertex starts with in-degree zero, its probability of being connected is always zero. To solve this problem, Price used k_0+k as the initial in-degree and k_0 is a constant. In his mathematical development, $k_0=1$ which means the paper cited by itself as the first citation[30]. In the result, if the in-degree is just 1, it will not show on the result.

At the same time, if the out-degree is 0 it has not linked to other pages.

The pseudo code of connection shown below is similar with the code described in section 2.4. AdjMatrix is used to record the relationship between nodes, SelectANodeRandomly is to select a node in the tree with the same probability, Path() is to get the nodes of the shortest path from source node to the destination node and “degree” is a Hashmap to collect the in-degree. Comparing with using matrix to collect value, the Hashmap not only can enhance the efficiency but also express the value clearly. In this program, every connection of a node will be recorded while in the original version the matrix can only show whether the node is linked or not.

0. Initial the adjmatrix and degree (the hidden tree has been generated)

```
1. for (int i = 0; i < N; i++)
2.   act = activity;
3.   while (act > 0)
4.     rnd = random(0,1);
5.     if (rnd < act)
6.       int nodenum = SelectANodeRandomly();
7.       ArrayList<Node> path = Path(i + 1, nodenum);
8.       int d = degree.get(nodenum);
9.       degree.put(nodenum, d + 1);
a.     AdjMatrix[i][nodenum - 1] = AdjMatrix[nodenum - 1][i] =
       AdjMatrix[i][nodenum - 1]+1;
b.     for (int j = 0; j < path.size(); j++)
c.       AdjMatrix[i][path.get(j).id - 1] = AdjMatrix[path.get(j).id - 1][i] =
       AdjMatrix[i][path.get(j).id - 1]+1;
d.     end for
e.   end if
f.   act = act - 1;
g. end while
h. end for
```

The result of this program is to produce a Hashmap with all of the in-degree of nodes. The parameters of this model are the number of nodes N, the number of branches n which decides the number of subpages of a page and the probability of connection “activity” which controls how many nodes that a node can connect to. In Fig.12, Fig.13, Fig.14, it is to test the effect of three parameters in websites separately.

To test the affection of the number of branches (the number of child nodes of each node), the experiments choose N=5000, activity = 5, and the n = 2,15,30 to get Fig.12.

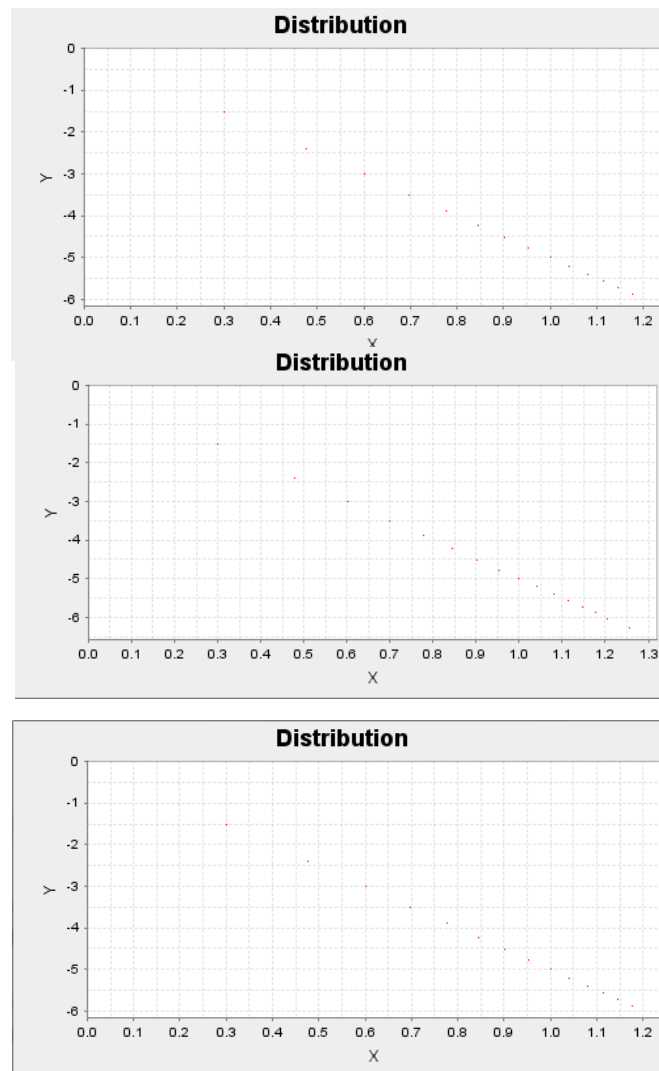
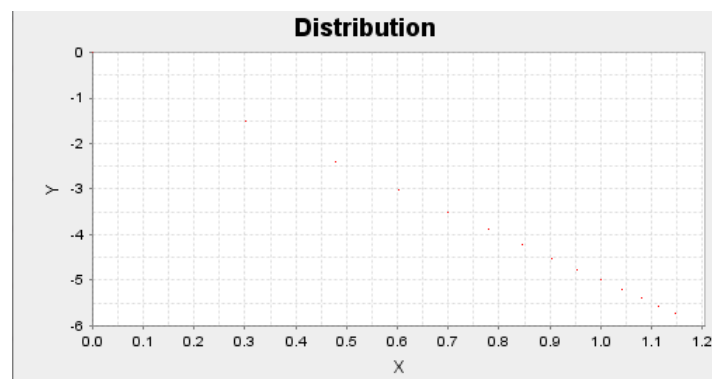


Fig.12

To test the affection of the number of nodes, the experiments choose activity = 5, $n=15$, and the $N = 1000, 3000, 5000$ to get Fig.13.



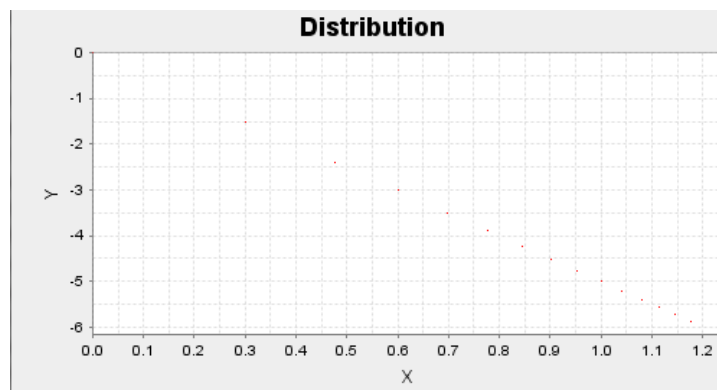
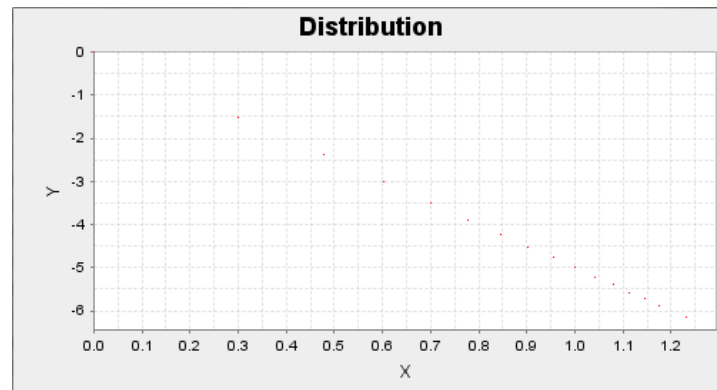
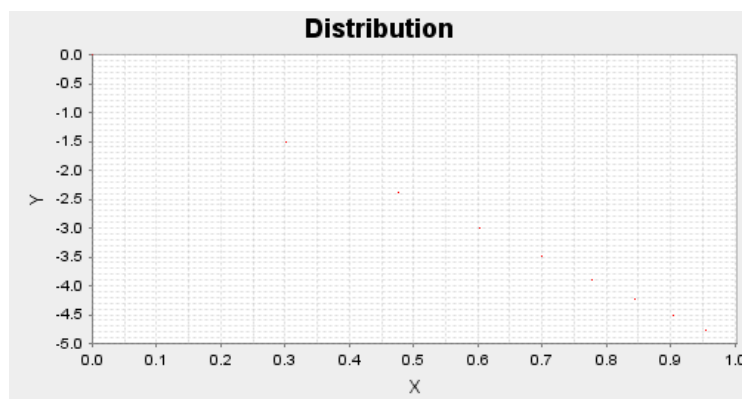


Fig.13

To test the affection of the activity, the experiments choose $n = 15$, $N = 5000$, and the activity = 2, 5, 10 to get Fig.14.



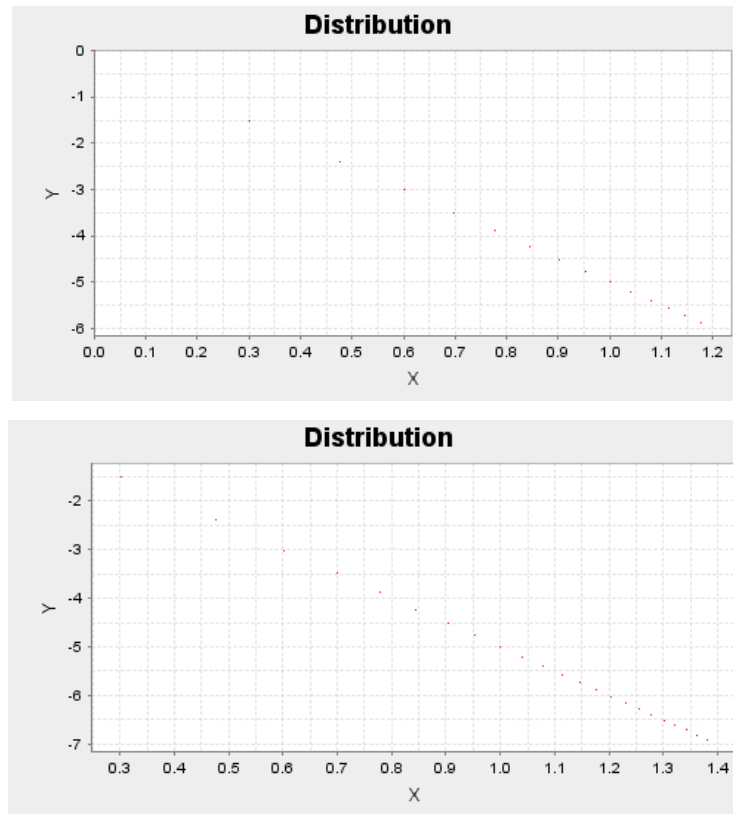


Fig.14

In these figures, it is easy to know that the degree of nodes follows the power law distribution in different conditions. The increase of parameters' value is the same as to increase the scale of the website. In this case, the points in the scattergram still follow the power law distribution and the position is similar, which means the γ is not affected by the growth of the website. Therefore, hidden tree model is proved to be a possible explanation of emergence of scaling in the World Wide Web.

Compared with the BA model, the hidden tree model avoids the challenge of global information hypothesis. In the BA model, to create a connection, the node has to know which node has higher connectivity. In this program, the in-degree of a node is added, because the node is pointed or it is on the path between source and destination node, and the destination node is selected randomly. In a way, this method and Alexei's has something similar.

In Alexei's model, the "adding+walking" is a process that a new vertex joins into the network and learns "knowledge". The new vertex links to the vertices pointing to the selected vertex in a probability. In Alexei's dissertation, he made an example for "adding+walking" that is a process of the author citing other papers and getting more similar papers through the references of the cited papers. In the hidden tree model, it has a similar "walking" step. The tree structure is similar with the cognitive structure of human. When a node points to another node, it has a high probability to link with its superior nodes.

3.3 The variants of the Hidden Tree model

3.3.1 “the node with role”

The first variant of hidden tree is to add the roles into the model. The main idea is to change the rule of connection to make the network more real: only leaf nodes can be connected. In last section, the nodes are “equal” which means they follows the same regulation to run. However, in the real network, each web-page may play different roles. For example, the main page includes the hyperlinks of the most popular content and rare heavy text. In BBC website, both the main page, www.bbc.co.uk, and the second level web page, www.bbc.co.uk/news, are the aggregates of links. Especially in second level web page, the links may point to the content web-pages or the index of themes. Therefore, in website, each webpage has its personal function.

This variant is just a very simple way to know what will happen when increasing the complexity of the network. An important feature of this variant is although the function to select leaf node is random; the whole hidden tree mode is not growing absolute randomly. The growth of the World Wide Web always contain many implicit rules, the different connecting probability between content pages and index pages is a key rule. In the real World Wide Web, the index web page of a website is not likely to be removed and the number of index pages does not change a lot in a long period. Adding or deleting index page may affect the structure of the website but there has no same problem for content web pages. This is a special point in hidden tree model, changing a role of node has no big difference with rebuilding a new tree model.

In this program, after building hidden tree by using the algorithm in section 3.2, the algorithm of creating connection needs to use the function `SelectALeafNodeRandomly()` to replace the `SelectANodeRandomly()`. When building the tree, the identity of numbers of leaves are recorded in an array list, so all of the destination nodes are selected from this list randomly. The parameters of this program are also the number of nodes N , the number of branches n and the probability of connection.

To compare with the result in the last section, the value still choose $N=5000$, activity = 5, and the $n = 2,15,30$ to get Fig.15.

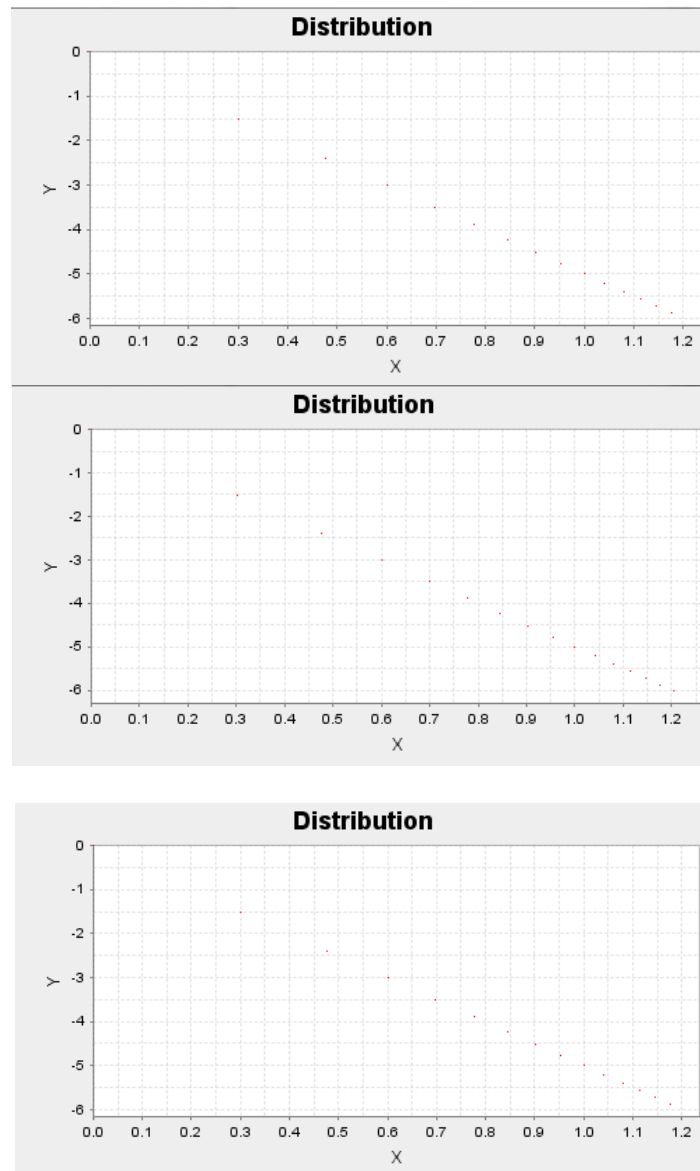


Fig.15

To test the affection of the number of nodes, the experiments choose activity = 5, $n=15$, and the $N = 1000, 3000, 5000$ to get Fig.16.

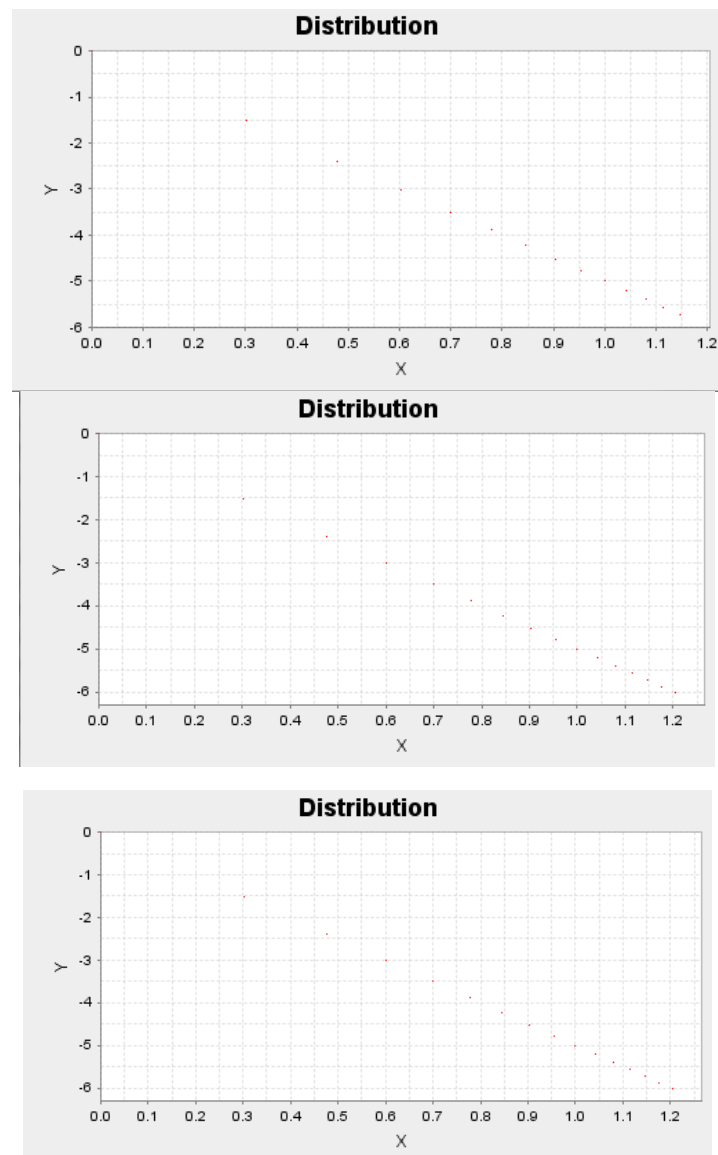
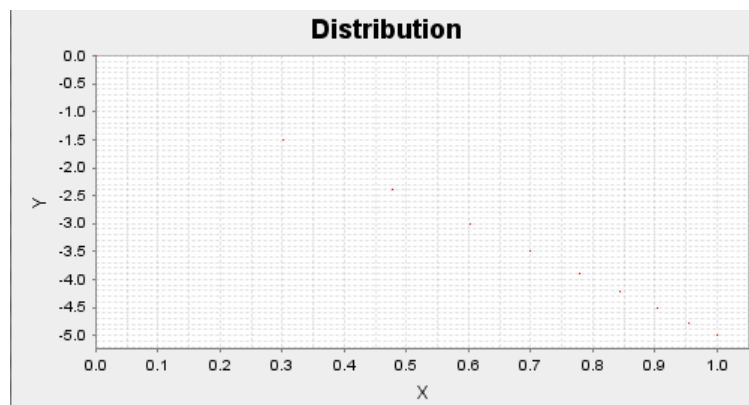


Fig.16

To test the affection of the activity, the experiments choose $n = 15$, $N = 5000$, and the activity = 2, 5, 10 to get Fig.17



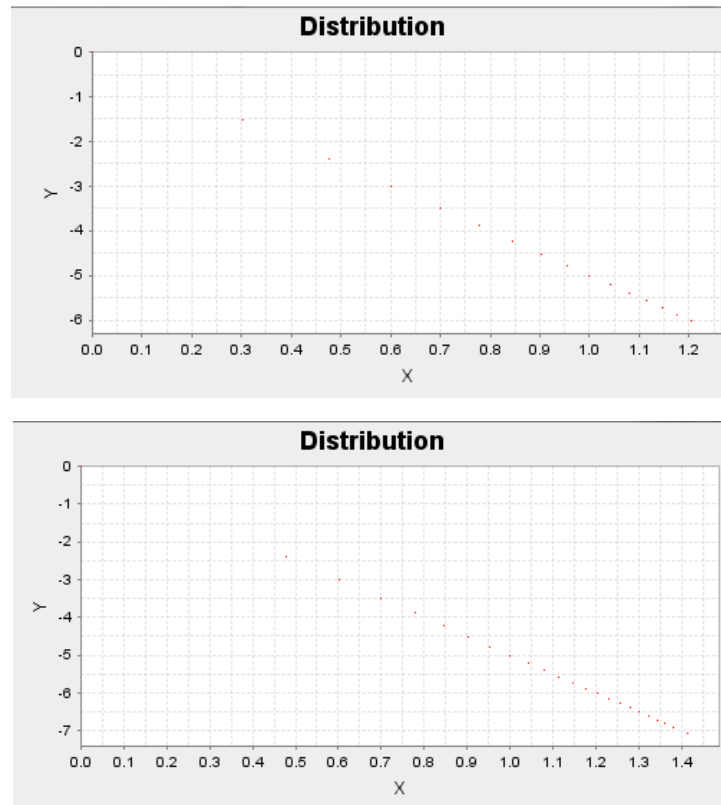


Fig.17

The result of the figures proves that in different conditions, the in-degree of nodes follows the power law distribution, so this variant of the hidden tree model can also explain the scaling in the World Wide Web. And there is an assumption which is the emergence of many special features of World Wide Web because of the irregular tree structure of the website.

3.3.2 Random connecting probability

To simplify the model, Bojing designed the node that has the same probability to be connected. This feature is the same as the ER model in which every vertex has the same connecting probability in a random graph. In the BA model, the probability is affected by the preferential attachment feature. However, Adamic and Huberman have discussed the problems of the preferential attachment. They believe each vertex in the network should have individual growth rate. So this variant model is to generate a tree with random connecting probability to explore the potential of the Hidden Tree model.

In the variant, the basic rules are:

- a. A full n-tree (every parent node has same number of children)
- b. Each node has a random probability to be connected
- c. The tree is static
- d. Each node can be connected

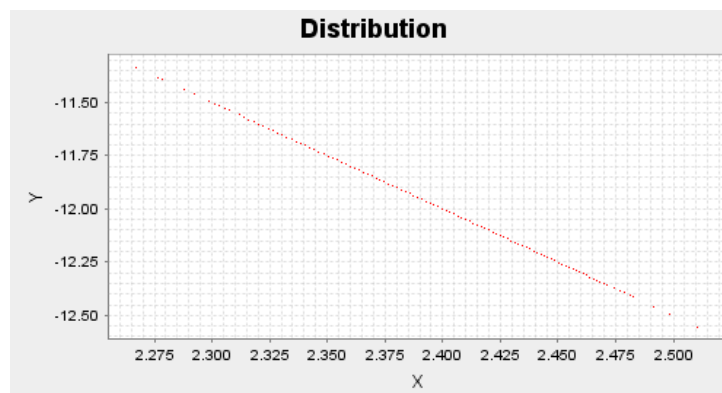
In this variant, because the value of activity is random, the program needs to make a change, and the input parameters just include the number of nodes and the number of branches. The pseudo code of the program is:

```

0. Initial the adjmatrix and degree (the hidden tree has been generated)
1. for (int i = 0; i < N; i++)
2.   act=randominteger(0,500);
3.   while (act > 0)
4.     rnd = random(0,1);
5.     if (rnd < act)
6.       int nodenum = SelectANodeRandomly();
7.       ArrayList<Node> path = Path(i + 1, nodenum);
8.       int d = degree.get(nodenum);
9.       degree.put(nodenum, d + 1)
a.     AdjMatrix[i][nodenum - 1] = AdjMatrix[nodenum - 1][i] =
       AdjMatrix[i][nodenum - 1]+1;
b.     for (int j = 0; j < path.size(); j++)
c.       AdjMatrix[i][path.get(j).id - 1] = AdjMatrix[path.get(j).id - 1][i] =
       AdjMatrix[i][path.get(j).id - 1]+1;
d.     End for
e.   End if
f.   act = act - 1;
g. End while
h. End for

```

This program is to generate a Hashmap which records the in-degree value of each node and a matrix showing the connection between nodes. The parameter acting in the code is to select an integer randomly to control the activity of nodes, such that each node has an individual activity value. The result is in Fig.18 and Fig.19. In this variant model, the first group is to test the effect of n and another group is to test the number of nodes. Thus, in first group, the N=5000 and n = 2, 30, while in the second group, the n = 15 and N = 1000, 5000.



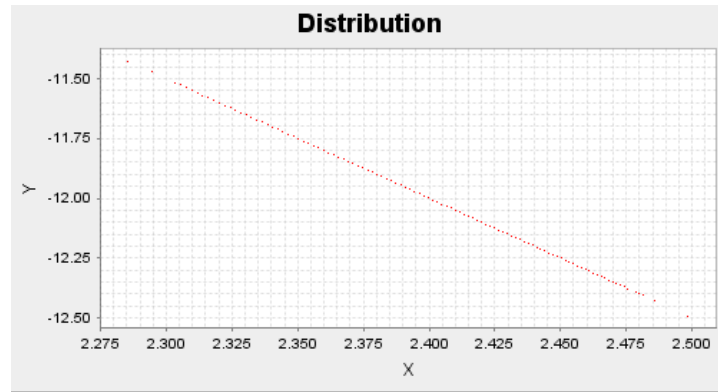


Fig.18.

In Fig.18, the scattergram follows the power law distribution, but there are some gaps at the beginning and end. This phenomenon may be the effect of the tree structure, because in the tree model, the nodes in the top have more probability to be connected through the path, which increases the difference of nodes' degree.

In Fig.19, the gram also follows the power law distribution, so the connecting probability of nodes does not affect the model in explaining the emergence of scaling in World Wide Web.

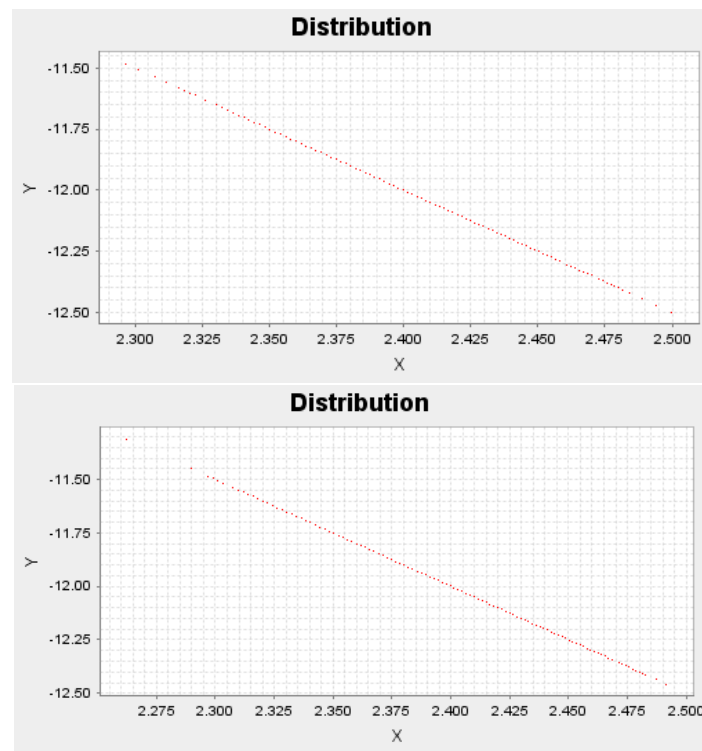


Fig.19

Based on the result of random connecting probability model, the variant can generate the power law distribution. Therefore, in the tree structure, the connecting probability does not play a key role to decide the distribution. Comparing with the BA model, the probability is not controlled by the preferential attachment theory, so the challenge for the correlation

between age and degree can be solved. In fact, the hidden tree structure does not have the disadvantage of preferential attachment, but it still has some features of it. This point will be analysed in section 3.4.

3.3.3 Random number of branches

In the previous experiments, the hidden tree model is always a full N-tree, but this ideal structure is not possible to appear in the real World Wide Web. In a website, the content of each sub-index is different. For example, in a news website, the number of pages about business news is 10,000, but the number about sports is just 100. In this case, the branches of the tree have different scales. The part for business is a thick sub-tree while the sub-tree for sports only has a few nodes. This situation may be changed with the growth of the web-site. The huge number of web-pages tends the difference to zero.

In this variant, the connection algorithm is the same as before. The nodes can connect with others randomly. But the algorithm of building the hidden tree is changed. In the following pseudo code, the input value can control the number of nodes, the activity of connection and the maximum branches number. When building the tree, the program select a random integer number as the branch number. `nodealt` is to collect the nodes not added into the tree and the `dealt` is to collect the tree nodes. `count` is to record the available child nodes number of nodes, `rcount` is to record the available number of node in the level and `rankcount` is the number of all nodes in the level. `Rank` is to record the level of node and `degree` is a hashmap to record the degree of each node.

```
0. initialnodealt();
1. int count = n - 1;
2. int rcount = 0;
3. int rankcount = 1;
4. int parentid = 0;
5. int level = 1;
6. int i = 0;
7. int j = 0;
8. int a=0;
9. while nodealt is not empty
a.     j++;
b.     if (rcount == rankcount)
c.         rcount = 0;
d.         rankcount = n * rankcount;
e.         level += 1;
f.     end if
g.     if (count == n)
h.         parentid++;
i.         a=random (0,n)
j.         count = a;
k.         rankcount=rankcount-a;
l.     end if
```

```

m.   Rank.put(j, level);
n.   degree.put(j, 1);
o.   dealt.add(nodealt.get(i));
p.   nodealt.remove(i);
q.   dealt.get(j - 1).parentnodeid = parentid;
r.   count++;
s.   rcount++;
t.   end while

```

The input value is the same as section 3.2, so the results are shown in Fig.20, Fig.21 and Fig.22.

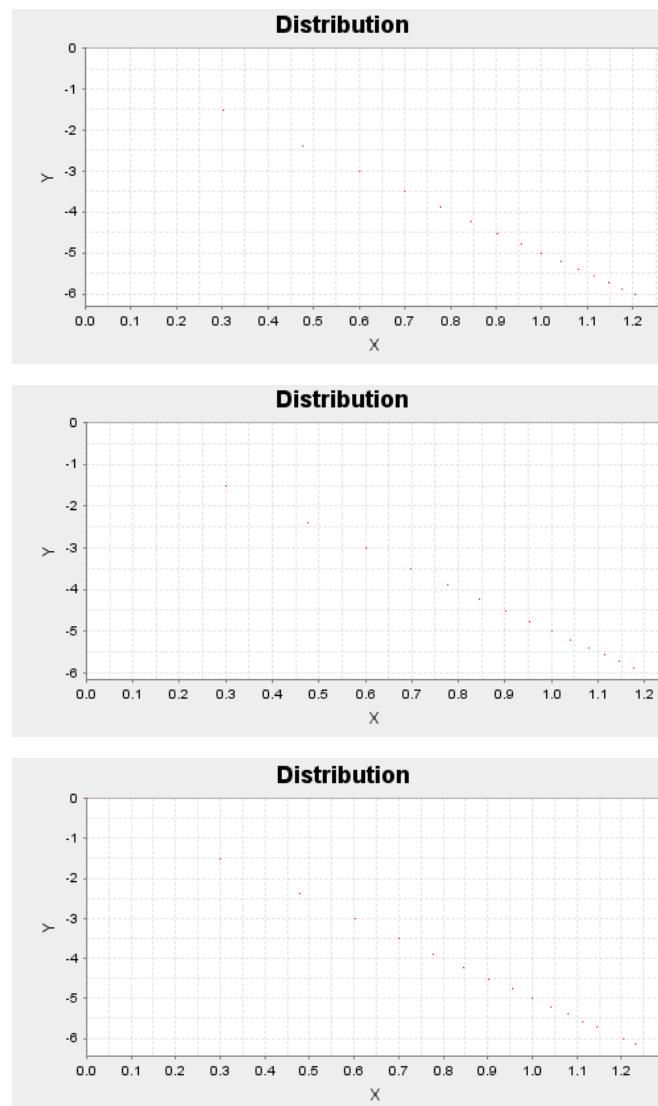


Fig.20

In fig.20, the degree of nodes follows the power law distribution like the result in section 3.2. Thus, it can get the same conclusion as section 3.2 which is with the increase of nodes number the value of exponent γ does not change.

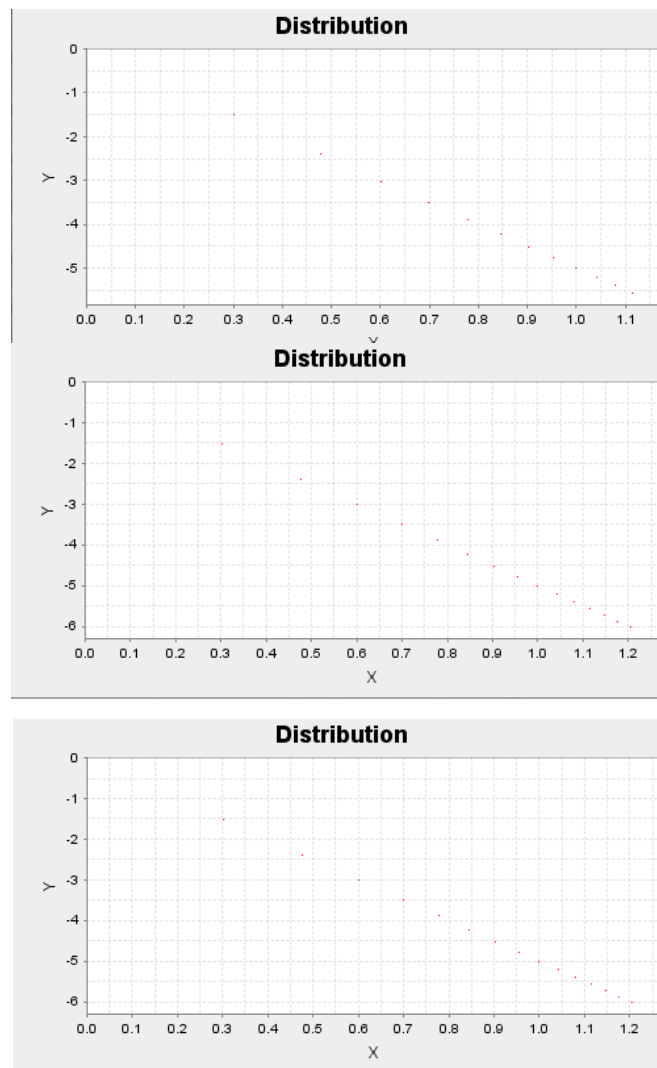
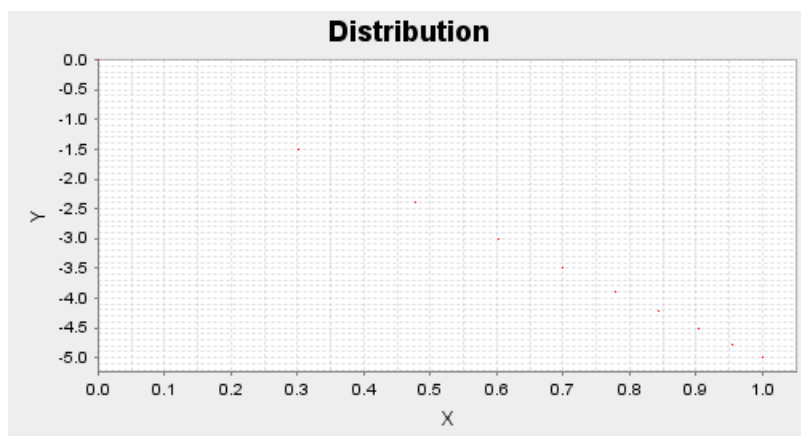


Fig.21

In Fig.21, with the change of n , the position of points in the graph is still linear, so the model also satisfies the power law distribution.



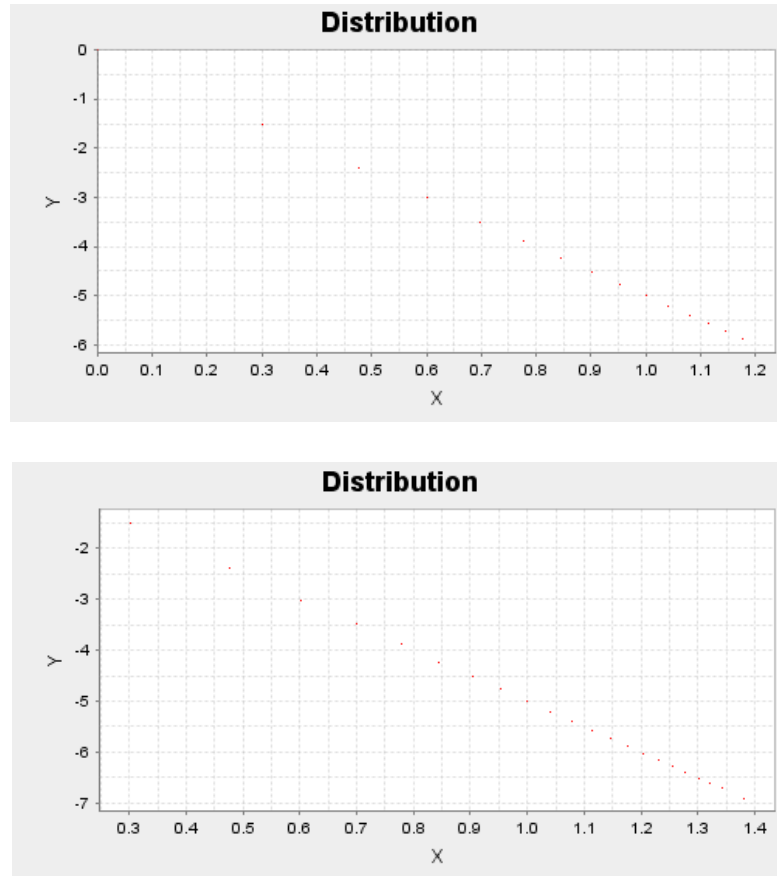


Fig.22

In Fig.22, the in-degree still follows the power law distribution.

Based on the result above, in different input parameters, the degree of nodes always follows the power law distribution. Thus, this variant can explain the distribution phenomenon in different scales of the website.

3.3.4 Hidden Tree model with random branch number and role

This variant is the combination of section 3.3.1 and 3.3.3. In the following sections, the models will combine the previous model in order to map the complexity of the World Wide Web. In sections 3.3.1 and 3.3.3, the results prove the variants can explain the power law distribution. In theory, the exponent γ does not change with the change of parameter, so after combining the two models, the exponent should not change. Thus, the variant should also follow the power law distribution. However, in the real web, there is no perfect theory to explain the distribution of degree. The distribution is affected by many implicit reasons. Just for Hidden Tree model, this new theory still has a large unknown potential to be explored. This variant is to test whether the features will be kept with the increasing of complexity.

The algorithms of this variant are using a new building tree algorithm to build a tree and

record the leaf node, collect all of the leaves' id into a hashset and connect nodes in 3.3.1's algorithm. Before building the tree, each node has a new parameter: leaf. Each node added into the tree is a leaf until it has a child node. The pseudo code of building a tree and searching for a leaf is shown below:

Tree:

```

0. Initialize the nodealt and dealt
1. while nodealt is not empty
2.     if (rcount == rankcount)
3.         rcount = 0;
4.         rankcount = n * rankcount;
5.         level += 1;
6.     end if
7.     if (count == n)
8.         parentid++;
9.         a=random (0,n)
a.         count = a;
b.         rankcount=rankcount-a;
c.     end if
d.     Rank.put(j, level);
e.     degree.put(j, 1);
f.     dealt.add(nodealt.get(i));
g.     nodealt.remove(i);
h.     dealt.get(j - 1).parentnodeid = parentid;
i.     If the node has parent
j.         dealt.get(parentid - 1).leaf = false;
k.         count++;
l.         rcount++;
m. end while

```

Collecting Leaves:

```

0. Initialize LeafNodeId
1. LeafNodeId = new HashSet<Integer>();
2. for (int i = 0; i < dealt.size(); i++)
3.     if the node is leaf
4.         LeafNodeId.add(dealt.get(i).id);
5.     End if
6. End for

```

The input value of the parameter is to test the number of nodes and the activity. Thus, in group one, the input value is N = 1000,5000 and activity = 10. The result is in Fig.23:

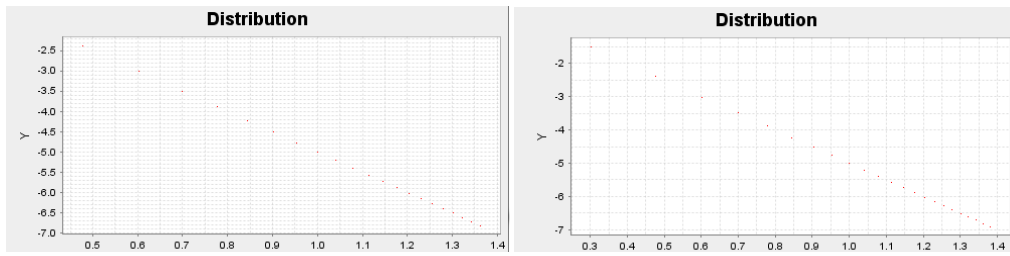


Fig.23

The second group is $N = 5000$ while activity = 10,20. The result is in Fig.24:

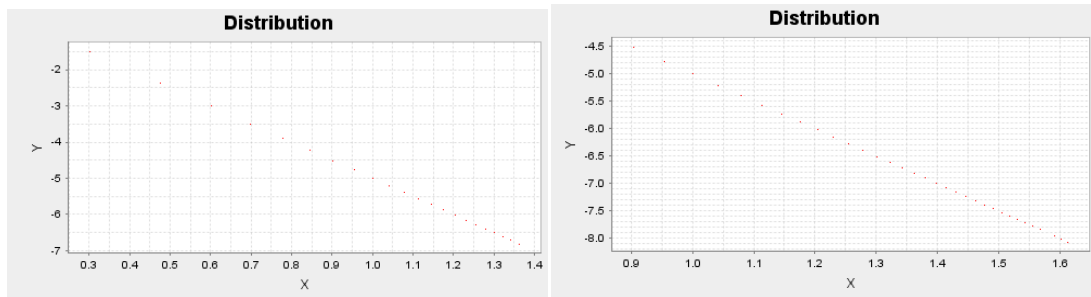


Fig.24

In the figures, all of the results follow the power law distribution, so the hidden tree model seems able to play a bigger role in mapping complex networks.

3.3.5 Hidden Tree With Role And Random Activity

This model is to test if the nodes of hidden tree have a personal role and the random connecting probability. The basic rule of the model is:

- It is a full n -tree
- Only leaves can be connected
- Each node has random probability to be connected

In the first group, to test the affection of number of nodes, the input values are activity = 5, $n=15$, and the $N = 1000, 3000, 5000$ to get Fig.25.

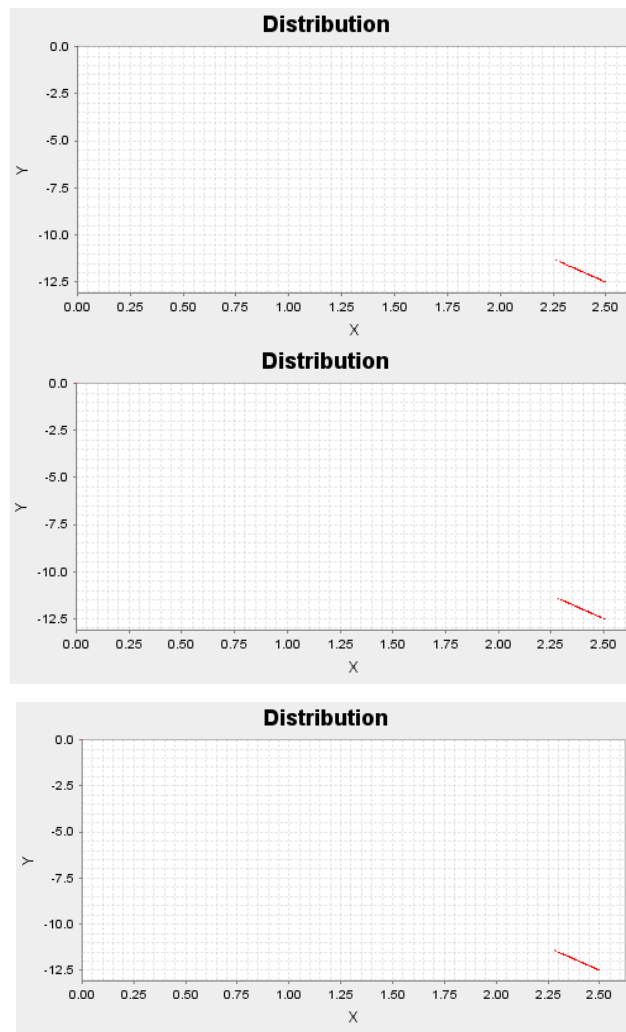
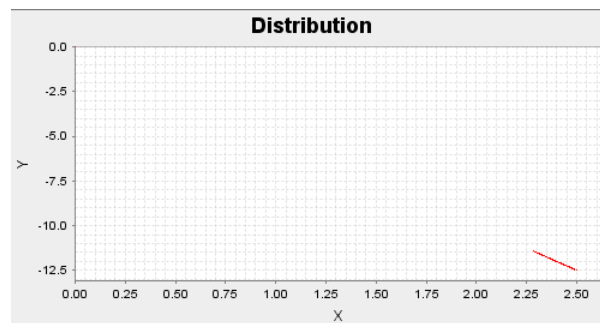


Fig.25

To test the affection of the n-tree, the value still chooses $N=5000$, activity = 5, and the $n = 2,15,30$ to get Fig.26.



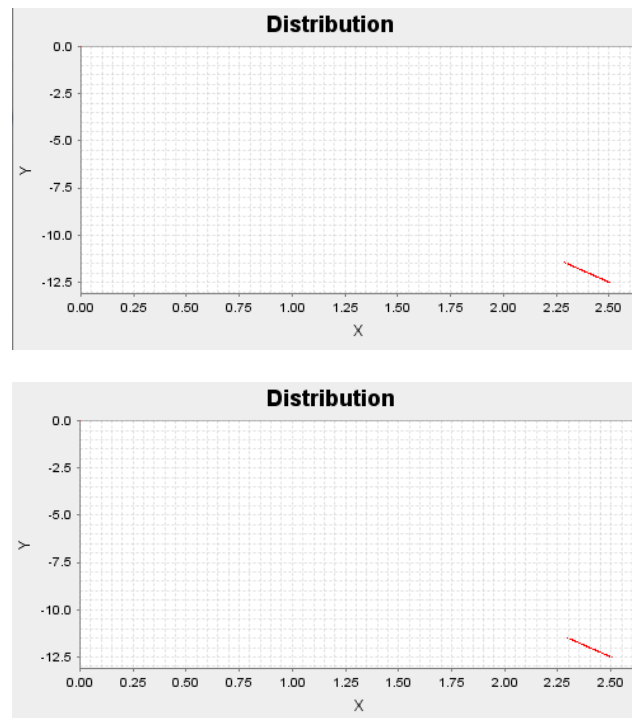
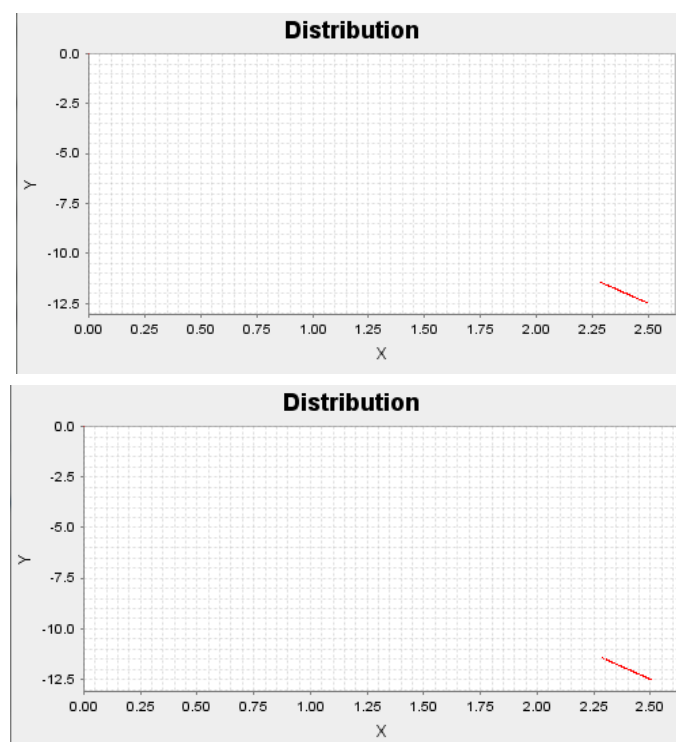


Fig.26

To test the affection of the activity, the experiments choose $n = 15$, $N = 5000$, and the activity = 2, 5, 10 to get Fig.27



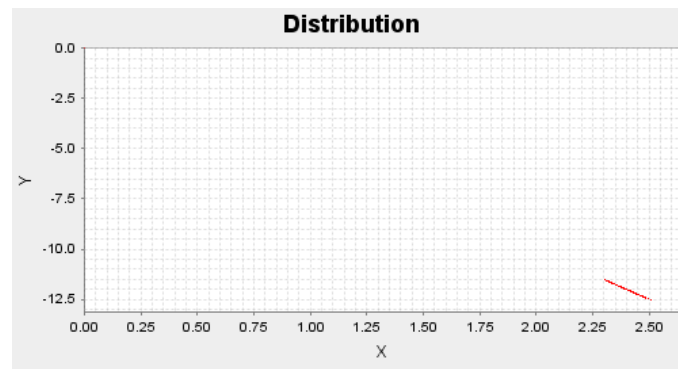


Fig.27

In these result, the degree of node also follows the power law distribution. This again confirms the feasibility of hidden tree theory in complex Web.

3.3.6 The tree model with random activity, structure and role

This variant is to combine the above variants together, and it is the last tree model in this project. This variant includes all the possible structures of a tree. In the World Wide Web, if the website is a tree structure, it may be found in this model. Obviously, most websites are not standard tree structures. In the website, the existence of a hyperlink leads to multifarious structures, but it does not affect the accuracy of this model. The generated tree model is a main part of the website. The nodes with hyperlinks just can shorten the limit for number of paths rather than change the whole structure into an irregular graph. And the challenge of these structures will be analysed in the next section.

In the result of this model (Fig.28, 29, 30), the complex structure does not change the power law distribution.

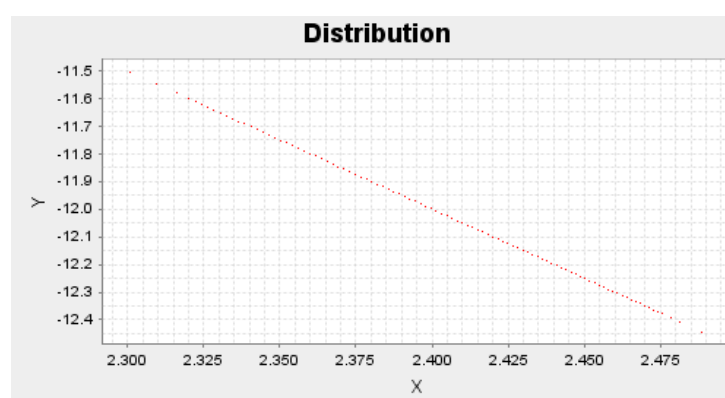


Fig.28

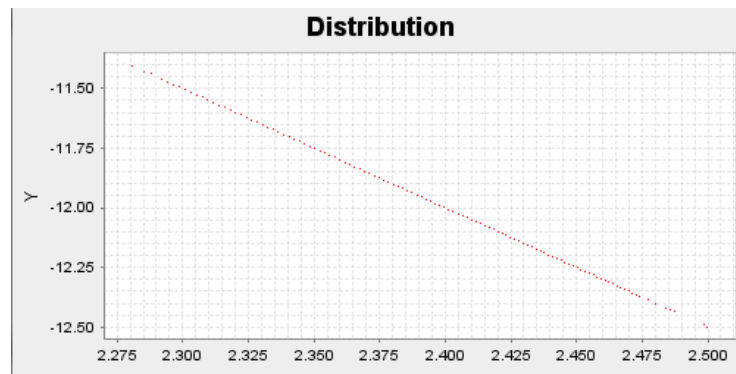


Fig.29

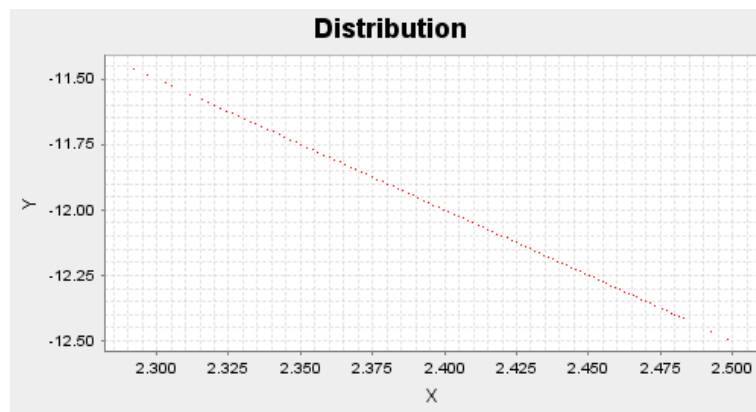


Fig.30

3.3.7 Hidden Graph

The topic of structure of a website has been discussed many times in the previous sections. For a structure of a network, the vertices can belong to a few networks or websites, or from a different angle, the web is constructed of different networks. Does the hidden tree structure still explain the emergence of scaling in the World Wide Web? In this section, the variant is not a tree structure but a directed random graph constructed by two hidden trees. A sample graph is shown in Fig.31 below. The left part can be connected by the right one but it is a single way path. This model can be understood as two websites connecting with each other or one complex website. For example, in a website, the left part may be the content of business and the right one is for politics. The top nodes are at the same level which may be a main index. And the leaf shared by two parts is news about both business and politics. In the theory of BA model, a contentious idea is all of the vertices belong to a single component, but in the real web, there are many components [12]. This model could be a new and simple component in the web. An assumption can be proposed if there are millions of similar components connecting together, the model can grow into a real World Wide Web.

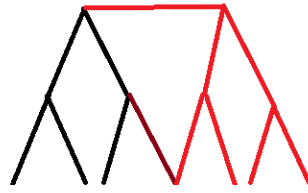


Fig.31

In this program, the algorithm of hidden tree is the simplest algorithm like 3.2. The algorithm of connection is the pseudo code below:

```

1.   for (int i = 0; i < (N1+N2); i++)
2.       act = activity;
3.       while (act > 0)
4.           rnd = Rd.nextFloat();
5.           if (rnd < act)
6.               int nodenum = SelectANodeRandomly();
7.               if (i < N && nodenum > N)
8.                   act = act - 1;
9.               end if
a.           else
b.               recorddegree(nodenum);
c.               act = act - 1;
d.           end else
e.       end if
f.       end while
g.   end for

```

The program uses SelectANodeRandomly to select a node randomly. If the source node is from the left part, the function will just select the destination node in the left part. For the node in the right part, it can link to any node in the model. This program just records the in-degree of all nodes in a Hashmap degree by function recorddegree(). Like the original hidden tree program, all of the nodes on the shortest path from source node to destination node are linked by source node, so the in-degrees of them are increased.

The input parameters are two groups including the number of nodes, the number of branches and the probability of connection.

To test the affection of N, the input value is n=15, activity = 15 and N=1000,2000,5000. The other group uses the same input value. The result is Fig.32.

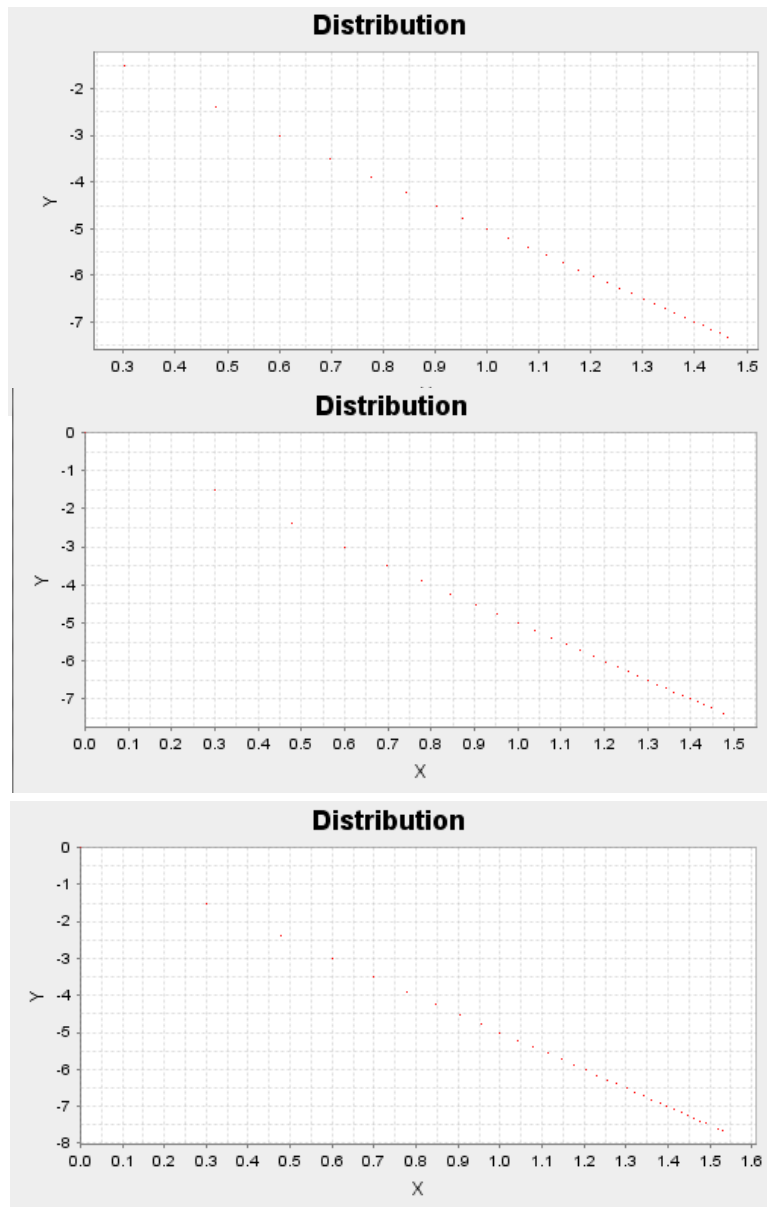


Fig.32

To test the affection of n , the input value of two groups are activity = 15 and 30, $N=1000,5000$ and $n = 2,10,30$. The result is Fig.33

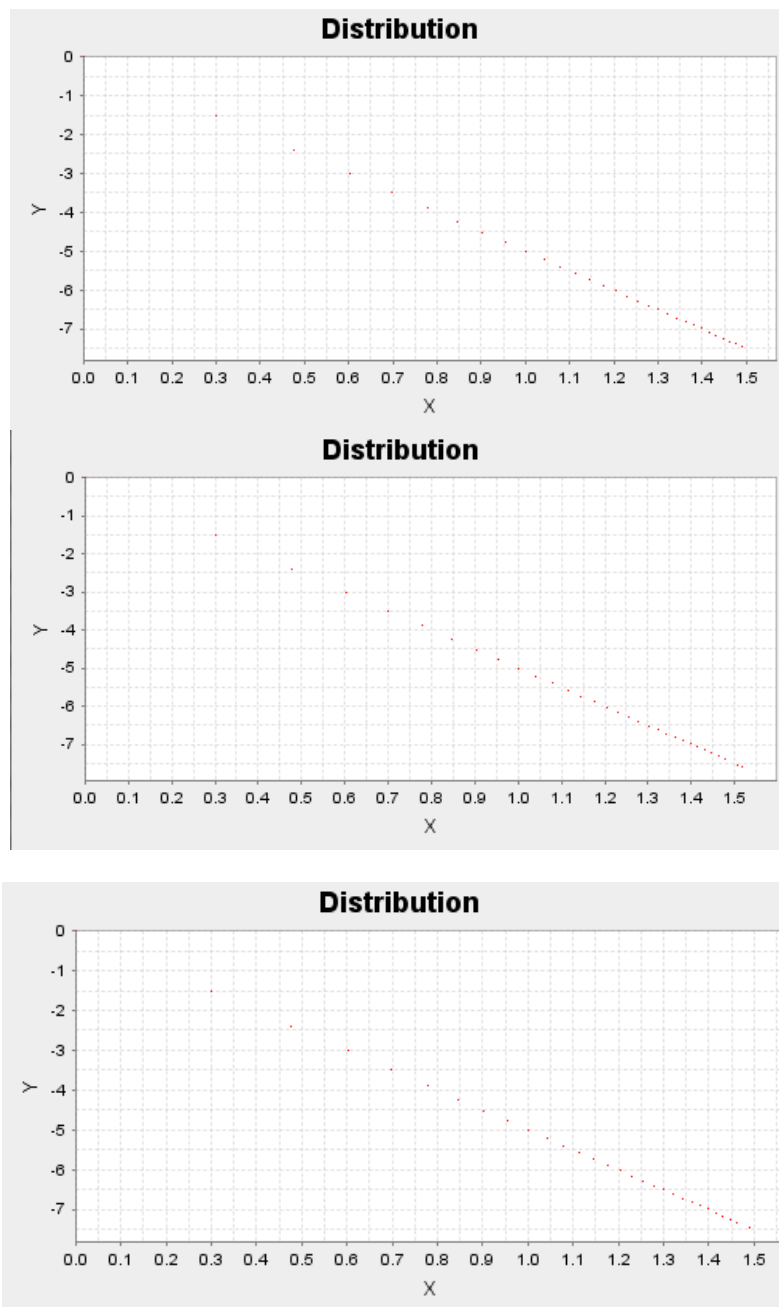


Fig.33

To test the affection of activity, the input values are $N=1000,5000$, $n=15$ and activity = 10,20,30. The result is Fig.34

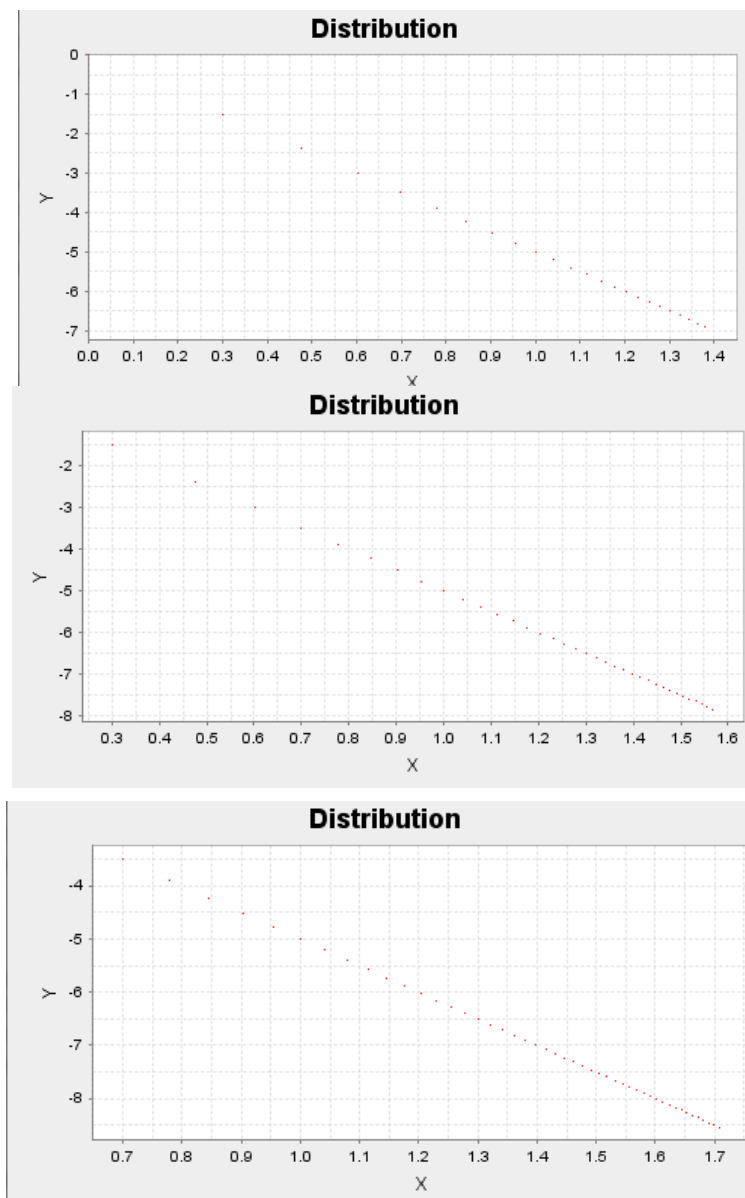


Fig.34

In the result, it is easy to know the model follows power law distribution. Therefore, this result can prove the structure of the network does not have to be a tree structure. The structure just needs to be similar to a tree. In other words, the network is a “forest” with many different “trees”. These trees may be the large websites or a combination of a few websites. Thus, the hidden tree theory can explain the general situation.

3.4 Wealth model

The appearance of a fast development website cannot be explained in the BA model, such as the fast-rising Facebook and Google. The rising of a website reflects the growth of wealth or popularity of the website, so a website can get more connections with others. In the World

Wide Web, the wealth of website is the number of visits. If a website is more famous, it may be connected by more web pages. In the BA model, the preferential attachment theory, an expression of the Matthew Effect, decides the new vertex is impossible to get a large number of connections when there has any other old vertex with high connectivity. Although Barabasi and Albert explained the scaling in WWW does not have a direct relationship with the age of website, the main idea of the BA model still expresses a Matthew Effect.

In BA model, the amount of links of a network leads to the unfair result. For example, the news on the BBC website may be linked by millions of web pages every day, but a blog of a small website is impossible to be connected by a lot of websites. This is, of course, a general phenomenon is a new website is not likely to have the same probability to be visited like a well-known website. That is why people believe in preferential attachment. However, this theory is just suited for common websites. For Facebook or Google, the BA model cannot give a good explanation.

In the real world, the age and scale can enhance the popularity of a website, which can give some positive assistance for growth. For example, a book of a famous author must get more attention than a rookie, even though the content of a young author is perhaps more wonderful. But, the development of a website is still dependent on many other reasons, such as the good management and nice structure of the company, the competition between websites and the financial market. The research of the success of a website needs different fields of knowledge. For example, the emergence of Amazon leads to a new project about the Long Tail Effect. In the BA model, the model can get the whole information of a network, so it can give more connection to a vertex with large numbers of links but not a developing vertex. This project focuses on generating a model which can accept the appearance of these fast-rising websites.

3.4.1 Matthew Effect

The Matthew Effect [35] is a phenomenon of the wealthy people getting wealthier while the poor people get poorer. It is not only a phenomenon in finance but also very important in computer, biology and physics areas. This theory is used to explain the emergence of unbounded distribution of the mechanisms. In a mature society, Matthew Effect is a main feature while hierarchy is another one. However, the relationship of these two features is not clear, so it is hard to define the mechanism of wealth distribution in the network. An assumption is the power law distribution may be the result of both the Matthew Effect and hierarchical structure.

The Matthew Effect is a positive feedback which leads to the clustering in the society. After the research, the wealthy distribution may follow the power law distribution [34]. In maths, one of the mechanisms to generate power law distribution is the Matthew Effect. In WWW, preferential attachment proposed by Barabasi is an expression of the Matthew Effect. However, some doubts of preferential attachment in the beginning of this section, a growth of a website in WWW or a company in the financial market, may be dependent on the human activities or investment. Thus, in the complex network, there may be more reasons

related with the power law distribution.

An economist has proposed a viewpoint: if you share out all the money equally to every person, after a long period, most of the rich in the past will still be rich and the poor go back to being poor. This viewpoint is to describe the same beginning; the amount of wealth is not the only one reason to decide the growth. The rich have a better ability to create new wealth, so the growth of wealth is an exponent growth. However, this growth is not infinite. Yahoo was one of the top websites in the USA, but now, the price of Yahoo is less than a third of its peak.

In the real world, people cannot stay in the same beginning. In the Matthew Effect, the people starting earlier should get more benefits than the freshers. For example, if a person deposited a sum of money in a bank, after a hundred years, the interest of the money should be a large sum. However, this result is impossible to appear in the real world. The existing tax or inflation may give people less money than in the beginning. In WWW, Barabasi proved the age of a website does not affect the distribution because of the same reason. Thus, the Matthew Effect is not a good explanation for short term development.

The discussion above is not to go against the Matthew Effect but to explain there should be other mechanisms to combine with the Matthew Effect to explain the scaling in WWW. There is another important feature of society which is the hierarchical structure. In a company, the managers have higher income than employees. The owner of the company may get more profit than the total amount of managers and employees. The senior manager has more opportunities to get high-income projects. Thus, at the same time, the manager and employee create different economic value. Therefore, it can be assumed that the distribution of wealth is because of the structure of the company. If the result of unfair distribution is because of the structure, as a fact, it is impossible to avoid this result unless the structure is destroyed. Obviously, the structure of company cannot be changed. Thus, in this project, the model for this distribution is a hierarchical model, hidden tree model. Compared with the preferential attachment model, the hierarchical model may be a better explanation for power law distribution.

3.4.2 Models and Experiments

In the real world, the wealth has a relationship with the right. In other words, the level in the structure decides the wealthy. For example, in recent society, some businessmen also have large wealth, but in old society, the king is the richest man in the country. This example is based on a human having huge power, or a good beginning. What about a poor man? If a poor man starts to sell apples, at the beginning, he has to do all of the things by himself. After a time, if he gets enough funds, he can employ people to do the work such as selling or moving apples. If he still can get good profit, he can increase the number of employees to enlarge the scale of both. The cycling between employing and enlarging can bring more and more wealth to him. Thus, this poor man grows into a manager at a high level. In this example, the increase of funds depends on the enhancing of level but not the initial funding.

This cascade control mechanism has the features of the Matthew Effect, because in the process of growth, the enlarging brings more profit which makes the manager richer. However, this mechanism does not stop the poor man growing into a rich man. If this mechanism can explain the wealth distribution, this mechanism should also follow the power law distribution.

In WWW, the website is a tree structure. Within different websites, the topics of websites are different, which means the growth of the WWW links the pages in these websites together. In the previous section, a conclusion that the main structure of a network is a tree structure can lead to the degrees following the power law distribution to be proven. Therefore, if the wealth of a webpage is related with the level in the website, or the wealth of a website is related with the level in the WWW, the wealth distribution can give some help to analyse the growth of the website.

If there are N units which constitute a class I that is $I=I_1, I_2, I_3 \dots I_N$. Make the F stand for the personal wealth, so that $F= F_1, F_2 \dots F_N$. All of the units are in a tree structure and each unit has average number k units as sub-node. Therefore, a k -tree is created by using the algorithm of Hidden Tree.

For each transaction, if the value is 1, the value will be distributed within all of the units who participate in the transaction with a distribution function. For example, in a transaction, the team leader may get 50% of the profit and the team members divide the rest. The function uses the rank of unit to determine the value of wealth obtained by the unit. In the function, if the rank of unit is $\text{rank}(l_i)$, the units are shown as l_i and the distribution function $g(\text{randnum})$, the wealth distributed to a unit is::

$$F(l_i) = \frac{g(\text{rank}(l_i))}{\sum_{j=1,2,\dots,p} g(\text{rank}(l_j))}$$

If the transaction between units is a random and continuous activity, the wealth of each unit will follow the power law distribution. This conclusion is tested in the variant of hidden tree model with the following algorithm:

```

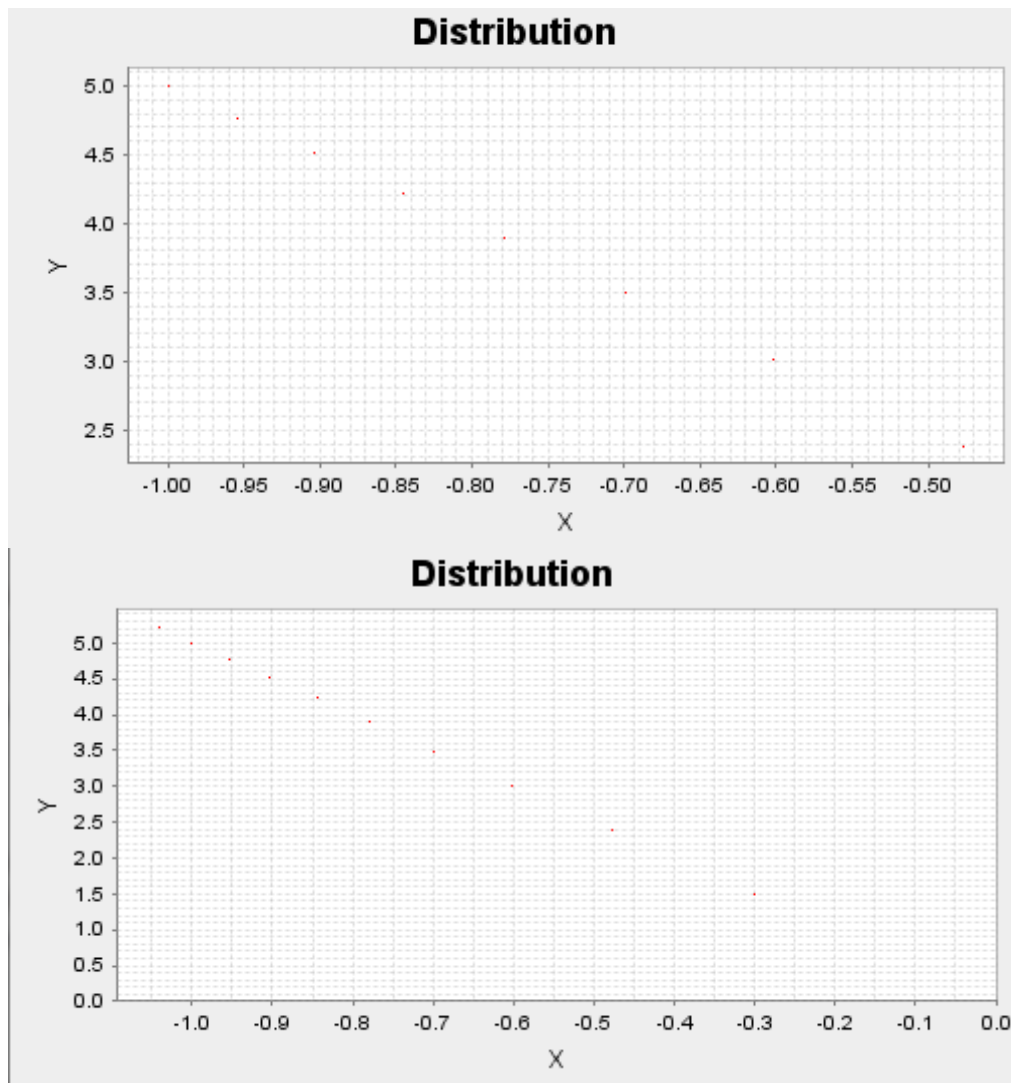
a. initialize F and hidden tree
b. for (int i = 0; i < N; i++)
c.     act = activity;
d.     while (act > 0)
e.         rnd = random(0,1);
f.         if (rnd < act)
g.             int nodenum = SelectANodeRandomly();
h.             ArrayList<Node> path = Path(i + 1, nodenum);
i.             float tmpF = Fpath(i + 1, path);
j.             Wealth.put(nodenum, tmpF);
k.         End if
l.         act = act - 1;
```

-
- m. End while
 - n. End for

In the algorithm, activity is the number of transactions for each unit, `SelectANodeRandomly()` is to select the destination unit randomly and path is to collect the units participating in the transaction. In the transaction, the path should be the shortest path from the source node to the destination node.

After building the model of wealth, there are four parameters that can affect the distribution. First of all, the number of units N which is the number of nodes in the tree; the second is the average sub-nodes number K , the third is the number of transactions for each unit A and the distribution function $g(\text{ranknum})$.

The first step is to test the effect of the scale of the network. If the $N = 1000, 2500, 5000$, and the other parameters are $K=5$, $A = 10$, $g(\text{ranknum})=1$, the result is in Fig.35.



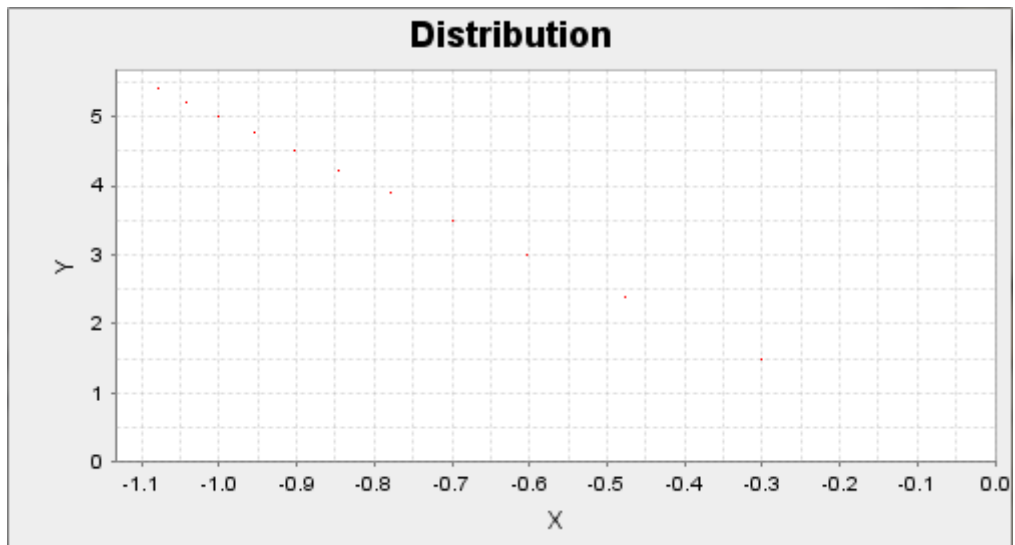
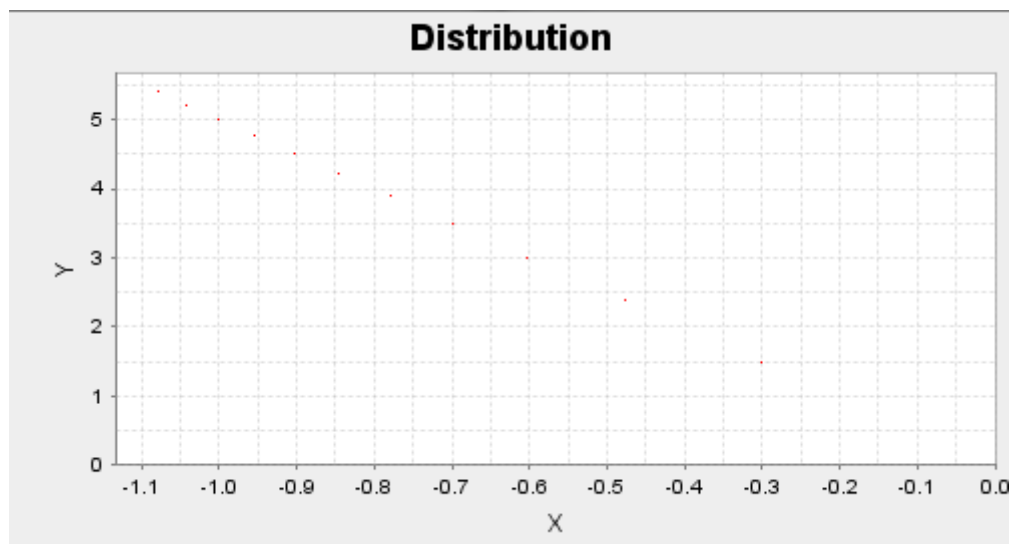


Fig.35

In fig.35, it can get a conclusion that the scaling of network does not affect the wealth distribution.

The second step is to test the effect of the number of sub-nodes for distribution. In this step, the N is 5000, A =10, $g(\text{ranknum})=1$ and $K = 5, 10, 20$. The result in Fig.36 also follows power law distribution. Therefore, it also states that the scale of network does not affect the distribution.



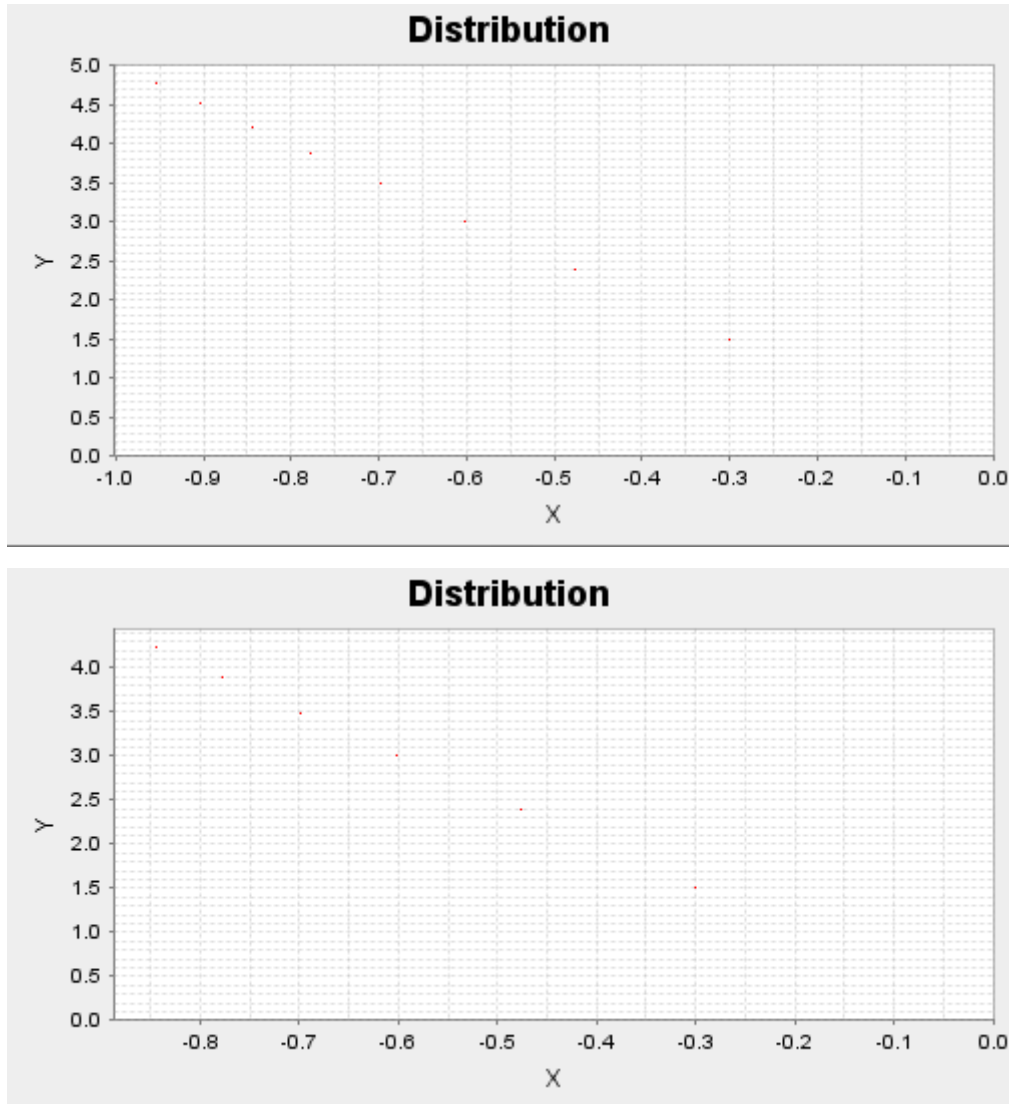


Fig.36

The third step is to discuss the effect of activity. For a node, the higher-activity can make more transactions so the wealth is higher. In this model, the activity is a constant value and every node has the same value. If the value of N is 5000, $g(\text{ranknum})$ is 1, $K = 10$ and activity = 10, 20, 30, the distribution is in Fig.37.

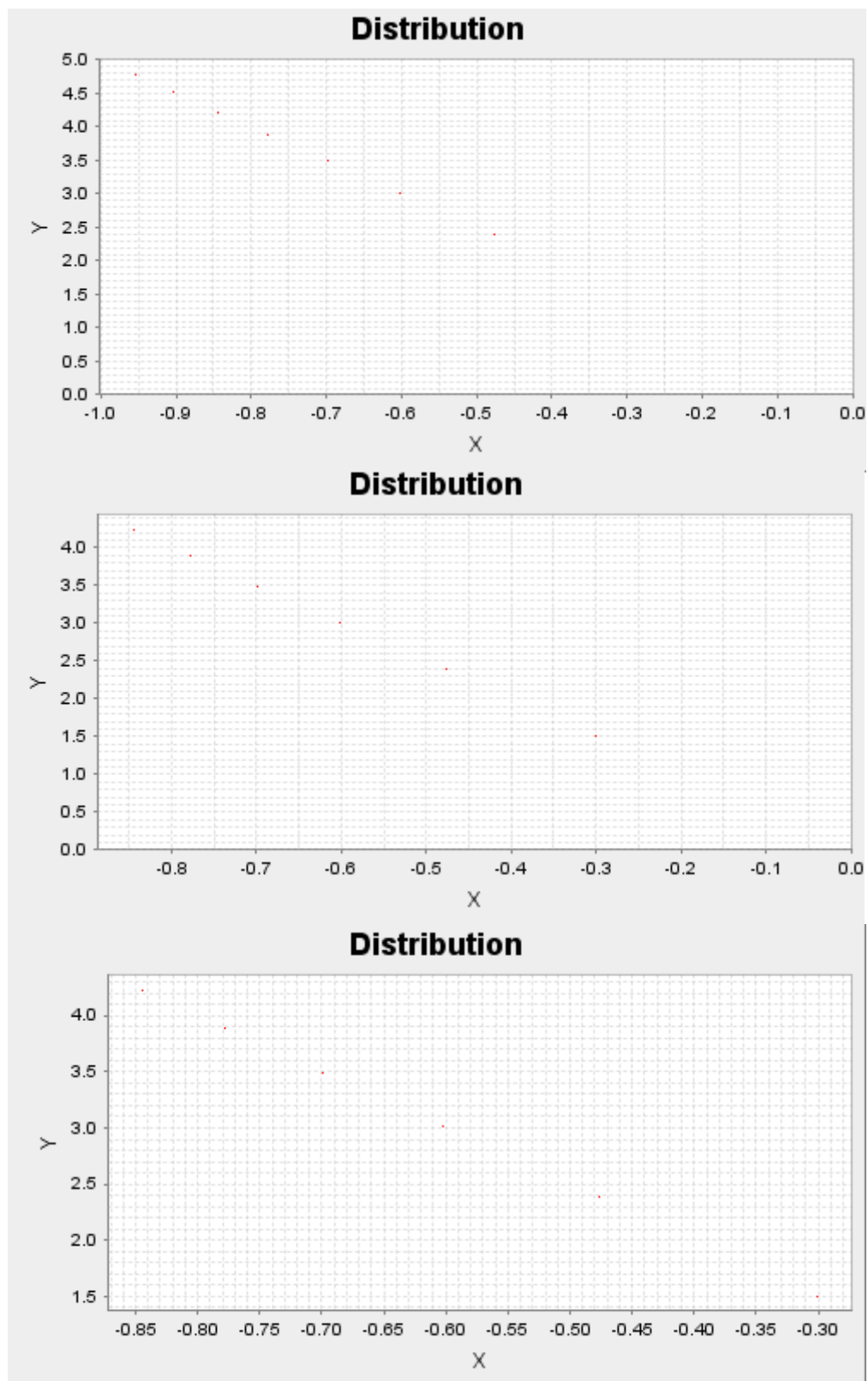


Fig.37

The last step is the analysis of distribution function. In the previous steps, the value of $g()$ is always 1, but for different ranks, the value may be different. To simplify the model, the

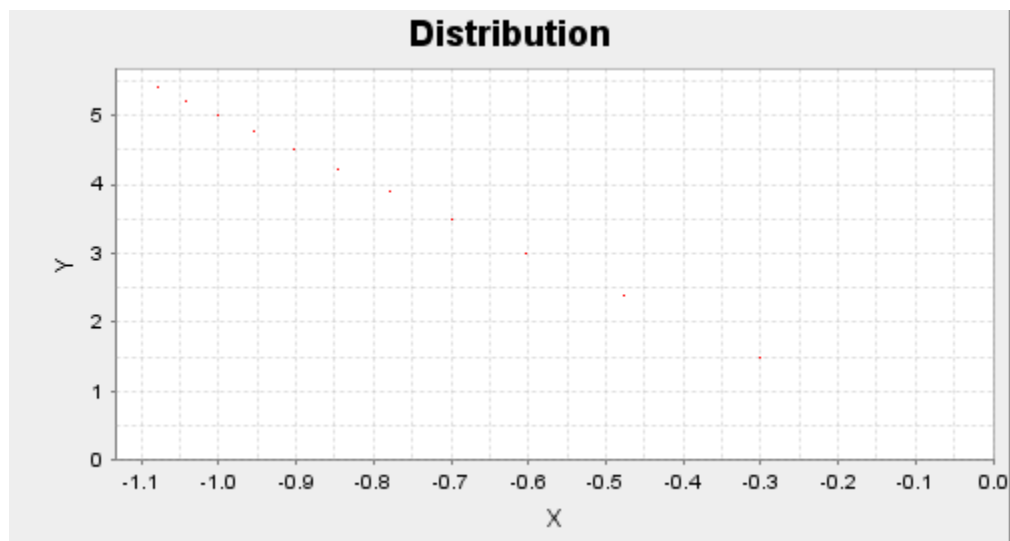
$g(\text{ranknum})$ is a simple polynomial which is $g(\text{ranknum}) = \text{ranknum}^t$. Thus, if $t=0$, the value of $g(\text{ranknum})$ is 1. The pseudo code of $g()$ is:

```

a.  if (k == 0)
b.      gli = 1;
c.  End if
d.  else
e.      if (k > 0)
f.          for (int i = 0; i < k; i++)
g.              gli = gli * gli;
h.          End for
i.      End if
j.      if (k < 0)
k.          k = 0 - k;
l.          for (int j = 0; j < k; j++)
m.              gli = gli * gli;
n.          End for
o.          gli = 1 / gli;
p.      End if
q.  End else

```

In this step, the parameters are $N=5000$, $K=5$, $A=10$ and $t = -2,1,2$. The distribution in Fig.38 is also a regular power law distribution.



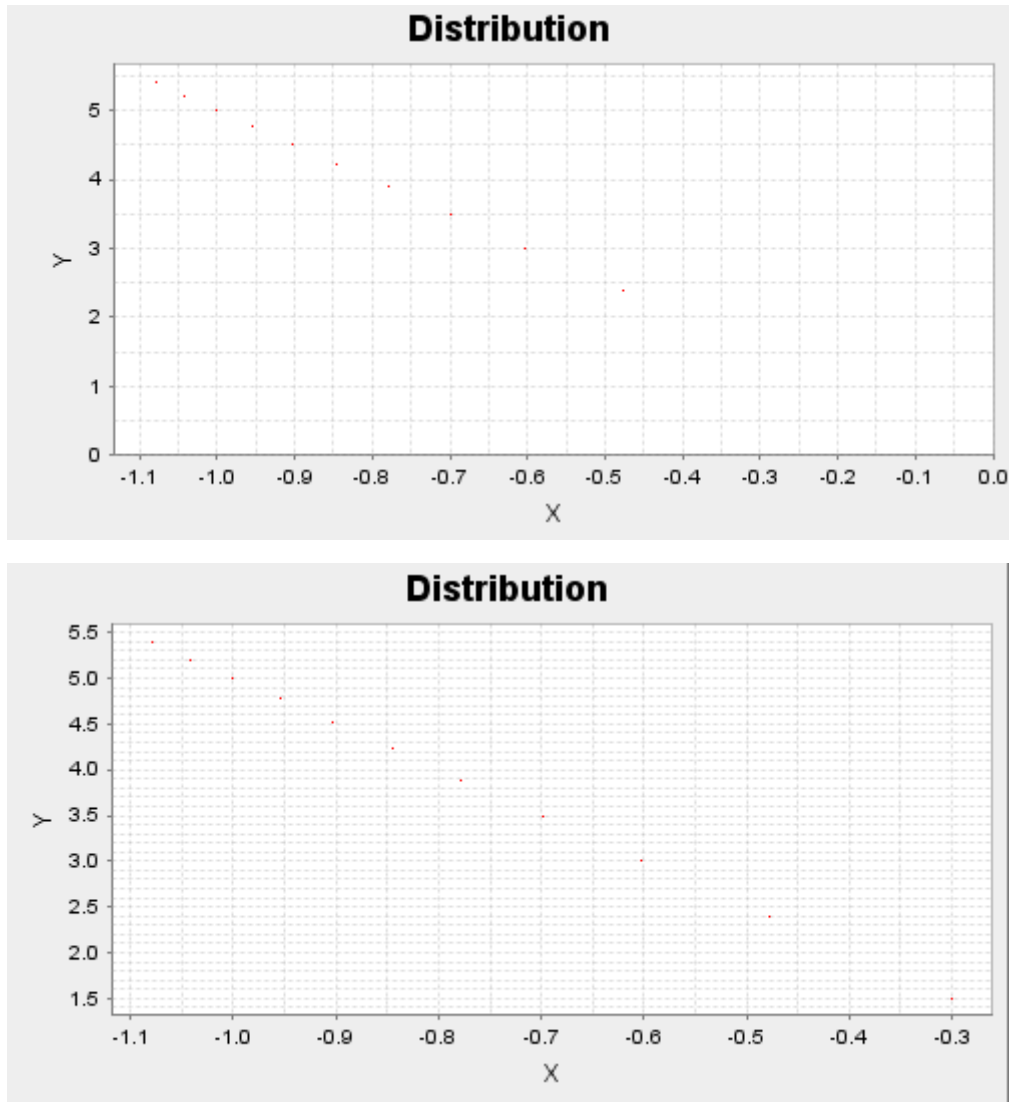


Fig.38

In these results, a clear viewpoint is in different scales, structures and distribution functions the wealth of a unit follows the power law distribution. Therefore, if the units are added into the tree structure, the power law distribution in the society can be explained.

3.4.3 Wealth model analysis

Preference is a very common phenomenon. In society, the effect of preference attachment seems more significant than in WWW. People pay attention to the successful people and prefer to establish relationships with them. However, in this theory, people will not pay attention to the growing people, so only grown successful people can enjoy the benefit of success. Therefore, this model is to apply the preference into the hierarchical organisation, in order to keep the feature of preferential attachment.

In the WWW, the structure of network leads to the unfair result. In other words, with the appearance of the fair result, there must be some problems with the structure. To negate

this idea, it must break the structure of the World Wide Web. As a network with a long history, the structure of the World Wide Web is tending towards stability. To get a fair result, the rationality is facing a big challenge. On the other hand, the unfair result gives some opportunities for new units to grow into a large group. The positive cycle between the rank and wealth can help a poor man move into the management level. This feature is impossible to appear in the BA model. Therefore, with the growth of websites, the appearance of Facebook or Google can be seen as a good example of the mutual promotion between wealth and scale. The theory of hidden tree model is more acceptable than the Matthew Effect.

3.5 Hidden Tree model analysis

If all of the vertices are seen as the agents, the connections between vertices are the votes from the source vertex to the destination vertex. In an undirected graph, the links are two-ways, so the in-degree is the evaluation from all other vertices and the out-degree is the evaluation to other vertices. Therefore, in WWW, the distribution of in-degrees is the distribution of accumulation of evaluations. And $f(X; x_i)$ represents the evaluation for x_i from the X vertices. This evaluation has different means in different angles. If the evaluation is the property of vertex, such as wealth, the distribution of wealth is the distribution of evaluating value; if the evaluation is the relationship between vertices, the degree distribution of a complex network is the distribution of evaluating value. After all, the distributions of degree or wealth are based on the same theory.

For the hidden tree model, no matter whether the simplest hidden tree model or the variants, such as only leaves as the destination node and the other nodes only connected when they are on the shortest path between source and destination nodes, or the two hidden trees constitute the hidden graph which limits the path between the trees, all obey the theory of hierarchy. For example, in a full binary-tree model with role, if a connection is made between two leaves, the probability of the root node being connected is $1/2$, because there are only 4 possible pairs for the leaves' position: A left B right, A right B left, A and B left or A and B right. If node A and B are in different parts, the root is on the shortest path. For the same reason, the connecting probability of nodes on the second level is $1/4$. Thus, the node on m level is 2^{-m} . For a full binary-tree, on level m , the number of nodes is 2^m . If every node has one opportunity to be a source node and there are M ranks in the network, the expected value of the root's degree is 2^{M-1} . For the nodes on m level, this value is 2^{M-m-1} . In this situation, the sum of expected value of all nodes on a level is always equal to 2^{M-1} and the expected value for each node on the same level is the same. Thus the probability of degree is:

The nodes' m which is the level of a node being exponential distribution, so:

$$p(m) \sim \frac{2^m}{\frac{2^{M+1} - 1}{2 - 1}} \approx 2^{-M+m-1}$$

The degree of node is related with level:

$$k = 2^{M-m}$$

Thus, the probability of degree:

$$p(k) \sim p(m) \frac{dm}{dk} = 2^{-M+m-1} \times \frac{1}{(2^{M-m})'} = -\frac{1}{2} k^{-2}$$

For the same reason, in n-tree:

$$p(m) \sim n^{-M+m-1}$$

$$k = \left(\frac{n-1}{n}\right)^{M-m}$$

So, it can derive $\gamma = 1 + \frac{\ln \frac{n}{n-1}}{\ln(n) - \ln(n-1)}$. When $n=2$, $\gamma = 2$.

This model gives a possible explanation of the distribution of the World Wide Web's degree. Using the structure of a website, the hidden tree model improves the Erdos-Renyi model to get a complex network with power law distribution. The degree of network is robust in different parameters' value. The model has the features of hierarchy, so it has some property of preference, but it does not have the same problems of preferential attachment mechanism. Thus, it can be used to explain some phenomena of a fast-rising website.

4 Summary

The project achieves two main aims. The first is the theory research of the emergence of scaling in the World Wide Web(WWW). The content of research includes 1) a study of two well-known network models, Erdos-Renyi(ER) model and Barabasi-Albert(BA) model, and the variants of the BA model, 2) an introduction of the World Wide Web and its growth 3) and the analysis of the new theory named Hidden Tree model with previous models. A network is structured by nodes and the links, while the World Wide Web is a huge network with web pages and hyperlinks. ER and BA models are well-known network models in history. The ER model supports that most networks are Poisson random models while the BA model explains networks following the power law distribution which has been accepted by most people. The key theory of the BA model is named preferential attachment. However, comparing with a simple network, the World Wide Web contains an enormous amount of pages and links and more complex changes. To explain the emergence of scaling in the World Wide Web, the original BA model has some difficult challenges, such as the global information hypothesis, the existence of linear preference and the emergence of new big internet companies [3]. The core model of this project, Hidden Tree model, is based on a very common structure of a website: a tree combined with the ER model to explain the emergence of scaling in the

WWW. At the same time, it can still explain some challenges of the BA model.

The second component of this project is to analyse the hidden tree theory and exploit the potential by comparing and combining it with existing network models. The results proved the Hidden Tree model can explain the emergency of scaling in WWW and the challenges of the BA model. The reasons affecting the results include the number of nodes, the structure of a tree, the probability of connection and the rule of connection. The program used these reasons as the parameters or added them into the algorithm in order to prove the networks based on the hidden tree structure follow the power law distribution and analyse the reasons for errors. The variants of the hidden tree model were designed to test whether the changing complexity of a model affects the features of the hidden tree. Finds indicated that the networks based on hidden tree model always follow the power law distribution, so this model can explain the emergency of scaling in many situations. For a specific variant model, the phenomenon of a fast big company in the WWW can get a simple explanation. Moreover, these variants combine the important features of the existing network model, such as Adamic and Huberman's model or Vazquez's model. The results prove the Hidden Tree model get around the challenges of the BA model.

5 Future work

After the experiments, the hidden tree model proposed a new angle to research the WWW. With the explaining power law distribution successful, there are more challenges and work to study.

The future work could be divided into two parts. One is for the research of the network, and the other is the hidden tree theory in other fields.

For the first part, the appearance of fast-rising websites needs more research. Although the hidden tree model can accept the existence of these websites, research needs more data to test. In this project, the experiment does not use the data about financial fields.

In this project, the generated network is a simple and small network. The World Wide Web is a complex graph, so the number of parameters is very large. However, a large complex system always has the problem of stability [36]. How to solve this problem is a reason to limit the development of the model. This project is just a beginning. It needs to do more variants to get more evaluation, such as changing the model from static to dynamic. Is there any mutation after changing the model? At the same time, the other fields' knowledge should be added into the model. For example, every day, the content of the website is renewed. Which kind of knowledge can explain this? How to add this into the model? All of these challenges may lead to an absolute opposite result.

In the second part, the challenges are from many other fields, such as sociology, biology and psychology. The hidden tree structure is an implicit structure in society, but this idea is based on common sense. Future work needs to analyse the source of this structure and how to

build a model for it is also a big challenge. In biology, there is also a hierarchical structure. A simple example is the food chain and people are positioned on top of the whole great pyramid of life. Is there any relationship between this hierarchical structure and the hidden tree? Can it be used to explain some phenomena in biology? In psychology, people have accepted this structure of society. This is an obedience activity of psychology. The psychological change process may be an essential reason for the source of the Hidden Tree.

References:

- [1] Bollobas, B.(1985). **Random Graphes** Academic Press, london.
- [2] Barabási,A. and Albert,R.(1999). **Emergence of scaling in random networks**. *Science* 286 509
- [3] Bojin,Z., Jianmin,W., Guisheng ,C., Jian,J. and Xianjun,S.(2011). **Hidden Tree Structure is a Key to the Emergence of Scaling in the World Wide Web**. *School of Software, Tsinghua University, Beijing 100084,China; School of Software, Tsinghua University, Beijing 100084,China; Institute of Chinese Electronic Engineering, Beijing 100840,China; Institute of Command and Technology of Equipment,Beijing 101416,China and Dept. of Computer Science, Central-China Normal University, Wuhan 430072,China*
- [4] Albert,R., Jeong,H. and Barabasi,A. (1999). **The diameter of the world wide web**. *NATURE* vol. 401
- [5] Huberman, B.A. and Adamic,L.A.(1999). **Growth dynamics of the World-Wide Web** *NATURE* vol. 401
- [6] Crow, E. L and Shimizu, K. (1988).**Lognormal Distributions: Theory and Applications** Dekker, New York
- [7] Berners-Lee,T. and Cailliau,R (1990). **WorldWideWeb: Proposal for a hypertexts Project**. Retrieved July 27, 2009.
- [8] www.W3C.org
- [9] Erdos, P. and Renyi, A. (1959). **On random graphs**. *Mathematicae* 6, 290–297.
- [10] Erdos, P. and Renyi, A., (1960). **On the evolution of random graphs**. *the Mathematical Institute of the Hungarian Academy of Sciences* 5, 17–61.
- [11] Erdos, P. and Renyi, A. (1961). **On the strength of connectedness of a random graph**. *Acta Mathematica Scientia Hungary* 12, 261–267.
- [12] Newman, M.E.J. (2003). **The structure and function of complex networks**. *SIAM Review*, Vol.45,No.2.pp.167-256
- [13] Merton, R.K (1968). **The Matthew Effect in Science**. *Science* 159 56
- [14] Willis, J.C. and Yule, G.U. (1922). Some statistics of evolution and geographical distribution in plants and animals, and their significance. *Nature* 109 177
- [15] Yule, G. U. (1925). **A Mathematical Theory of Evolution, based on the Conclusions of Dr.**

J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London, Ser. B* **213** (402–410): 21–87.

[16] Price, D. (1965). **Networks of Scientific Papers.** *Science* **149** (3683):510-515.

[17] Bornholdt, S. and Ebel, H. (2001). **World-Wide Web scaling exponent from Simon's 1955 model.** *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, Volume 64, Issue 3

[18] Krapivsky, P. L. and Redner, S. (2002), **A statistical physics perspective on Web growth,** *Computer Networks* **39**, 261–276.

[19] Krapivsky, P. L. and Redner, S. (2001), **Organization of growing random networks,** *Phys. Rev. E* **63**, 066123.

[20] Adamic, L.A. and Huberman, B.A. **Power-Law Distribution of the World Wide Web.** *Xerox Palo Alto Research Center 3333 Coyote Hill Road Palo Alto, CA 94304, USA*

[21] Vazquez, A. (2000). **Knowing a network by walking on it: emergence of scaling** *Abdus Salam International Center for Theoretical Physics Strada Costiera 11, P.O. Box 586, 34100 Trieste, Italy*

[22] Krapivsky, P.L., Redner, S. (2001). **Organization of growing random networks.** *Phys. Rev. E* **63** 066123

[23] Barabasi, A.L., Albert, R., Jeong, H., Bianconi, G. (2000): **Power-law distribution of the world wide web.** *Science* **287** 2115a

[24] Watts, D.J., Strogatz, S.H. (1998). **Collective dynamics of 'small-world' networks.** *Nature* **393**(6684) 440–442

[25] Bianconi, G., Barabasi, A.L. (2001). **Bose-einstein condensation in complex networks.** *Phys. Rev. Lett.* **86** 5632–5635

[26] Bianconi, G., Barabasi, A.L. (2001). **Competition and multiscaling in evolving networks.** *Europhys. Lett.* **54** 436–442

[27] Ravasz, E. and Barabasi, A. L. (2003). **Hierarchical organization in complex networks** *Phys. Rev. E* **67** 026112

[28] Zipf, G.K., (1949): **Human Behaviour and the Principle of Least Effort.** *Addison-Wesley, Cambridge, Massachusetts.*

[29] Bollobas, B., Riordan, O., Spencer, J., and Tusnady, G. (2001), **The degree sequence of a scale-free random graph process,** *Random Structures and Algorithms* **18**, 279– 290.

[30] Price, D. J. de S. (1976), **A general theory of bibliometric and other cumulative**

advantage processes, *J. Amer. Soc. In-form. Sci.* 27, 292–306.

[31] Slanina, F. and Kotrla, M. (1999). **Extremal Dynamics Model on Evolving Networks**. *Phys. Rev. Lett.* 83, 5587

[32] Christensen, K., Donangelo, R., Koiller, B. and Sneppen, K. (1998). **Evolution of Random Networks**. *Phys. Rev. Lett.* 81, 2380

[33] P. Bak and K. Sneppen, (1993): **Punctuated equilibrium and criticality in a simple model of evolution**. *Phys. Rev. Lett.* 71, 4083.

[34] Newman, M.E.J. (2005). **Power laws, pareto distributions and zipf's law**. *CONTEMP PHYS* 46 323–351

[35] Merton, R.K. (1968). **The matthew effect in science: The reward and communication systems of science are considered**. *Science* 159(3810) 56–63

[36] May, R.M. (1972). **Will a large complex system be stable?** *Nature* 238 413–414

[37] Holland, J. (1995) **Hidden Order: How Adaptation Builds Complexity**, *Massachusetts Institute of Technology*

[38] **Vazquez, A. (2002)**, Growing networks with local rules: Preferential attachment, clustering hierarchy and degree correlations, *Preprint cond-mat/0211528*.

[39] Rapoport, A. (1957). **Contribution to the theory of random and biased nets**. *Mathematical Biophysics* 19, 257–277.

[40] Solomonoff, R. and Rapoport, A. (1951). **Connectivity of random nets**, *Mathematical Biophysics* 13, 107–117.

[41] Butts, C.T.(2010). **Bernoulli Graph Bounds for General Random Graphs**. *University of California, Irvine*

[42] Watts, D. J. (1999). **Networks, dynamics, and the small world phenomenon**. *Am. J. Sociol.* 105, 493–592.

[43] Watts, D. J. (1999). **Small Worlds**. *Princeton University Press, Princeton*.

[44] Milgram, S. (1967). **The Small World Problem**. *Psychology Today*, 1(1), pp 60 – 67

[45] Monasson, R. (1999). **Diffusion, localization and dispersion relations on 'small-world' lattices**. *Eur. Phys. J. B* 12, 555–567.

[46] Newman, M. E. J. and Watts, D. J. (1999). **Renormalization group analysis of the small-world network model**. *Phys. Lett. A* 263, 341–346.

[47] Barrat, A. and Weigt, M. (2000). **On the properties of smallworld networks**. *Eur. Phys. J. B* 13, 547–560.

[48] Banks, D. L. and Carley, K. M. (1996). **Models for network evolution**, *Journal of Mathematical Sociology* 21, 173– 196.

[49] Davidsen, J., Ebel, H., and Bornholdt, S. (2002), **Emergence of a small world from local interactions: Modeling acquaintance networks**, *Phys. Rev. Lett.* 88, 128701.

Appendix:

Hidden Tree:

```
import java.util.*;

import org.jfree.ui.RefineryUtilities;
public class HiddenTree extends CommandWords {
    public ArrayList<Node> nodealt;
    public ArrayList<Node> dealt;
    public HashMap<Integer, Integer> Rank;
    public int N;
    // the children num of node;
    public int n;
    // the value in the matrix is the distance between each node;
    public int[][] Tree;
    public int[][] AdjMatrix;
    public float activity;
    private Parser HTP;
    private CommandWords htcw;
    private Index index;
    public HashMap<Integer, Integer> degree;
    private PRresult result;

    public HiddenTree() {
        nodealt = new ArrayList<Node>();
        dealt = new ArrayList<Node>();
        degree = new HashMap<Integer, Integer>();
        Rank = new HashMap<Integer, Integer>();
        HTP = new Parser();
        htcw = new CommandWords();
        index = new Index();
        index.HiddenTreeIndex();
        boolean finished = false;
        while (!finished) {
            Command NTC = HTP.getCommand();
            finished = processCommand(NTC);
        }
        System.out.println("Thank you for playing. Good bye.");
    }

    public void iniHiddenTree() {
        System.out.println("Please input the n,activity,N");
        n = 0;
        activity = 0.0f;
    }
}
```

```

    N = 0;
    BuildTree();
    TreeMatrix();
    AdjMatrix();
    HiddenTreeResult();
}

public void BuildTree() {
    HTP = new Parser();
    Command command = HTP.getHTCommandnum();
    n = command.getFirstWord();
    activity = command.getSWord();
    N = command.getThirdWord();
    initialnodealt();
    int count = n - 1;
    int rcount = 0;
    int rankcount = 1;
    int parentid = 0;
    int level = 1;
    int i = 0;
    int j = 0;
    while (!nodealt.isEmpty()) {
        j++;
        if (count == n) {
            parentid++;
            count = 0;
        }
        if (rcount == rankcount) {
            rcount = 0;
            rankcount = n * rankcount;
            level += 1;
        }
        Rank.put(j, level);
        degree.put(j, 1);
        dealt.add(nodealt.get(i));
        nodealt.remove(i);
        dealt.get(j - 1).parentnodeid = parentid;
        count++;
        rcount++;
    }
    // System.out.println(Rank);
}

public void initialnodealt() {

```

```

Node node;
for (int i = 1; i <= N; i++) {
    node = new Node();
    node.id = i;
    nodealt.add(node);
}
}

public void TreeMatrix() {
    Tree = new int[N][N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j <= i; j++) {
            if (i == j)
                Tree[i][j] = 0;
            else if (dealt.get(i).parentnodeid == (j + 1))
                Tree[i][j] = 1;
            else if (Rank.get(i + 1) == Rank.get(j + 1)) {
                if (dealt.get(i).parentnodeid ==
dealt.get(j).parentnodeid) {
                    Tree[i][j] = 2;
                } else
                    Tree[i][j] = 2 * (Rank.get(i + 1) - 1);
            } else {

                Tree[i][j] = Path(i + 1, j + 1).size()-1;
            }
            Tree[j][i] = Tree[i][j];
        }
    }
}

private ArrayList<Node> Path(int node1, int node2) {
    ArrayList<Node> Path1 = new ArrayList<Node>();
    ArrayList<Node> Path2 = new ArrayList<Node>();
    int i;
    Node Pnode1 = dealt.get(node1 - 1);
    Node Pnode2 = dealt.get(node2 - 1);
    for (i = 0; i < Rank.get(node1) - 1; i++) {
        Path1.add(Pnode1);
        Pnode1 = dealt.get(Pnode1.parentnodeid - 1);
    }
    for (i = 0; i < Rank.get(node2) - 1; i++) {
        Path2.add(Pnode2);
        Pnode2 = dealt.get(Pnode2.parentnodeid - 1);
    }
}

```

```

    }
    ArrayList<Node> Path = new ArrayList<Node>();
    int n1;
    int n2;
    int cn1 = 0;
    int cn2 = 0;
    boolean commonancestor = false;
    for (i = 0; i < Path2.size(); i++) {
        n2 = Path2.get(i).id;
        for (int j = 0; j < Path1.size(); j++) {
            n1 = Path1.get(j).id;
            if (n1 == n2) {
                commonancestor = true;
                cn1 = j;
                break;
            }
        }
        if (commonancestor) {
            cn2 = i;
            break;
        }
    }
    if (commonancestor) {
        for (i = 0; i < cn1+1; i++) {
            Path.add(Path1.get(i));
        }
        for (i = cn2-1; i >=0; i--) {
            Path.add(Path2.get(i));
        }
    } else {
        for (i = 0; i < Path1.size(); i++) {
            Path.add(Path1.get(i));
        }
        Path.add(dealt.get(0));
        for (i = Path2.size()-1; i >=0; i--) {
            Path.add(Path2.get(i));
        }
    }
    return Path;
}

public void AdjMatrix() {
    IniAdjMatrix();
}

```

```

    float act;
    Random rd = new Random();
    float rnd = 0;
    for (int i = 0; i < N; i++) {
        act = activity;
        while (act > 0) {
            //while (rnd == 0) {
                rnd = rd.nextFloat();
            //}
            if (rnd < act) {
                int nodenum = SelectANodeRandomly();
                ArrayList<Node> path = Path(i + 1, nodenum);
                int d = degree.get(nodenum);
                degree.put(nodenum, d + 1);
                AdjMatrix[i][nodenum - 1] = AdjMatrix[nodenum - 1][i]
= AdjMatrix[i][nodenum - 1]+1;
                for (int j = 0; j < path.size(); j++) {
                    AdjMatrix[i][path.get(j).id - 1] = AdjMatrix[path
                        .get(j).id - 1][i] =
AdjMatrix[i][path.get(j).id - 1]+1;
                }
            }
            act = act - 1;
        }
    }

    public void IniAdjMatrix() {
        AdjMatrix = new int[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                AdjMatrix[i][j] = 0;
            }
        }
    }

    public int SelectANodeRandomly() {
        Random rd = new Random();
        int nodenum = rd.nextInt(N);
        return nodenum + 1;
    }

    public void HiddenTreeResult() {

```

```

        result = new PRresult("HiddenTreeDistribution", degree);
        result.pack();
        RefineryUtilities.centerFrameOnScreen(result);
        result.setVisible(true);
    }

    private boolean processCommand(Command HTC) {
        boolean wantToQuit = false;

        if (HTC.isUnknown()) {
            System.out.println("I don't know what you mean...");
            return false;
        }
        if (HTC.getCommandWord().equals("1")) {
            iniHiddenTree();
        } else if (HTC.getCommandWord().equals("2")) {
            wantToQuit = htcw.quit(HTC);
        }

        return wantToQuit;
    }
}

```

Wealth:

```

public class Wealth extends CommandWords {
    public ArrayList<Node> nodealt;
    public ArrayList<Node> dealt;
    public HashMap<Integer, Integer> Rank;
    public int N;
    // the children num of node;
    public int n;
    // the value in the matrix is the distance between each node;
    public int k;
    public int[][] Tree;
    public float[][] AdjMatrix;
    public float activity;
    private Parser HTP;
    private CommandWords htcw;
    private Index index;
    public HashMap<Integer, Float> Wealth;
    private PRresultwealth result;
}

```

```

public Wealth() {
    nodealt = new ArrayList<Node>();
    dealt = new ArrayList<Node>();
    Wealth = new HashMap<Integer, Float>();
    Rank = new HashMap<Integer, Integer>();
    HTP = new Parser();
    htcw = new CommandWords();
    index = new Index();
    index.HiddenTreeIndex();
    boolean finished = false;
    while (!finished) {
        Command NTC = HTP.getCommand();
        finished = processCommand(NTC);
    }
    System.out.println("Thank you for playing. Good bye.");
}

```

```

public void iniHiddenTree() {
    System.out.println("Please input the n,activity,N,k");
    n = 0;
    activity = 0.0f;
    N = 0;
    k = 0;
    BuildTree();
    TreeMatrix();
    WealthCal();
    HiddenTreeResult();
}

```

```

public void BuildTree() {
    HTP = new Parser();
    Command command = HTP.getHTCommandnum();
    n = command.getFirstWord();
    activity = command.getSWord();
    N = command.getThirdWord();
    k = command.getFourthWord();
    initialnodealt();
    int count = n - 1;
    int rcount = 0;
    int rankcount = 1;
    int parentid = 0;
    int level = 1;
    int i = 0;
    int j = 0;
}

```

```

        while (!nodealt.isEmpty()) {
            j++;
            if (count == n) {
                parentid++;
                count = 0;
            }
            if (rcount == rankcount) {
                rcount = 0;
                rankcount = n * rankcount;
                level += 1;
            }
            Rank.put(j, level);
            Wealth.put(j, (float) 1);
            dealt.add(nodealt.get(i));
            nodealt.remove(i);
            dealt.get(j - 1).parentnodeid = parentid;
            count++;
            rcount++;
        }
        // System.out.println(Rank);
    }

    public void initialnodealt() {
        Node node;
        for (int i = 1; i <= N; i++) {
            node = new Node();
            node.id = i;
            nodealt.add(node);
        }
    }

    public void TreeMatrix() {
        Tree = new int[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j <= i; j++) {
                if (i == j)
                    Tree[i][j] = 0;
                else if (dealt.get(i).parentnodeid == (j + 1))
                    Tree[i][j] = 1;
                else if (Rank.get(i + 1) == Rank.get(j + 1)) {
                    if (dealt.get(i).parentnodeid ==
dealt.get(j).parentnodeid) {
                        Tree[i][j] = 2;
                    } else

```

```

        Tree[i][j] = 2 * (Rank.get(i + 1) - 1);
    } else {

        Tree[i][j] = Path(i + 1, j + 1).size() - 1;
    }
    Tree[j][i] = Tree[i][j];
}
}
}

```

```

private ArrayList<Node> Path(int node1, int node2) {
    ArrayList<Node> Path1 = new ArrayList<Node>();
    ArrayList<Node> Path2 = new ArrayList<Node>();
    int i;
    Node Pnode1 = dealt.get(node1 - 1);
    Node Pnode2 = dealt.get(node2 - 1);
    for (i = 0; i < Rank.get(node1) - 1; i++) {
        Path1.add(Pnode1);
        Pnode1 = dealt.get(Pnode1.parentnodeid - 1);
    }
    for (i = 0; i < Rank.get(node2) - 1; i++) {
        Path2.add(Pnode2);
        Pnode2 = dealt.get(Pnode2.parentnodeid - 1);
    }
    ArrayList<Node> Path = new ArrayList<Node>();
    int n1;
    int n2;
    int cn1 = 0;
    int cn2 = 0;
    boolean commonancestor = false;
    for (i = 0; i < Path2.size(); i++) {
        n2 = Path2.get(i).id;
        for (int j = 0; j < Path1.size(); j++) {
            n1 = Path1.get(j).id;
            if (n1 == n2) {
                commonancestor = true;
                cn1 = j;
                break;
            }
        }
    }
    if (commonancestor) {
        cn2 = i;
        break;
    }
}

```

```

    }
}
if (commonancestor) {
    for (i = 0; i < cn1 + 1; i++) {
        Path.add(Path1.get(i));
    }
    for (i = cn2 - 1; i >= 0; i--) {
        Path.add(Path2.get(i));
    }
} else {
    for (i = 0; i < Path1.size(); i++) {
        Path.add(Path1.get(i));
    }
    Path.add(dealt.get(0));
    for (i = Path2.size() - 1; i >= 0; i--) {
        Path.add(Path2.get(i));
    }
}
return Path;
}

public void WealthCal() {

    float act;
    Random rd = new Random();
    float rnd = 0;
    for (int i = 0; i < N; i++) {
        act = activity;
        while (act > 0) {
            rnd = rd.nextFloat();
            if (rnd < act) {
                int nodenum = SelectANodeRandomly();
                ArrayList<Node> path = Path(i + 1, nodenum);
                float tmpF = Fpath(i + 1, path);
                Wealth.put(nodenum, tmpF);

            }
            act = act - 1;
        }
    }
}

private float Fpath(int nodenum, ArrayList<Node> path) {
    float Fli;
    float ranki = FRank(nodenum);

```

```

        float rankj = 0;
        float sumrank = 0;
        for (int i = 0; i < path.size(); i++) {
            rankj = FRank(path.get(i).id);
            sumrank = sumrank + rankj;
        }
        Fli = ranki / sumrank;
        return Fli;
    }

    private float FRank(int nodenum) {
        int noderank = Rank.get(nodenum);
        float gli = (float) noderank;
        if (k == 0) {
            gli = 1;
        } else {
            if (k > 0) {
                for (int i = 0; i < k; i++) {
                    gli = gli * gli;
                }
            }
            if (k < 0) {
                k = 0 - k;
                for (int j = 0; j < k; j++) {
                    gli = gli * gli;
                }
                gli = 1 / gli;
            }
        }
        return gli;
    }

    public int SelectANodeRandomly() {
        Random rd = new Random();
        int nodenum = rd.nextInt(N);
        return nodenum + 1;
    }

    public void HiddenTreeResult() {

        result = new PRresultwealth("Wealth", Wealth);
        result.pack();
        RefineryUtilities.centerFrameOnScreen(result);
        result.setVisible(true);
    }

```