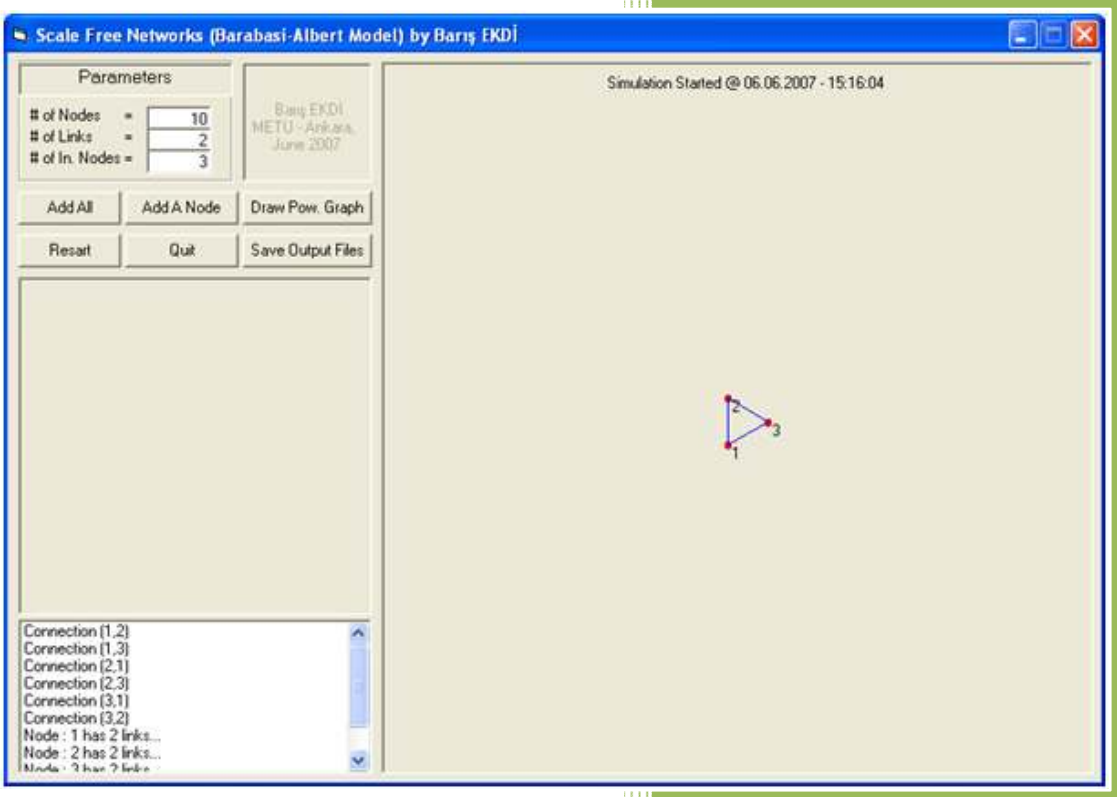
2007

# SCALE FREE NETWORKS




**Barış Ekdi**

6/20/2007

# SCALE FREE NETWORKS

# BARABASI & ALBERT MODEL

**by**

**BARIŞ EKDİ**

**Phd Candidate @ METU - STPS**

**Assignment No.V**

- **Basics of the Model & the Simulation**
- **Main Features of the Program & the Pseudo-Algorithm**
- **Output of the Simulation and Comments on the Simulation Results**

**Ankara, June 2007**

## 1. THE MODEL: Scale-free Networks and Barabási –Albert Model

Scale-free networks are complex networks because their structure and dynamics are independent of the system's size N, the number of nodes the system has. In other words, a network that is scale-free will have the same properties no matter what the number of its nodes is. Their most distinguishing characteristic is that their degree distribution follows a power-law relationship $P(k) \sim k^{-\gamma}$ where the coefficient γ may vary approximately from 2 to 3 for most real networks.

Scale free networks show certain properties which are very distinct in real networks. We often see a significant number of nodes in a network having a large number of links and a whole lot of nodes with very small number of links connected to them. A very good example is *Google* which is connected to literally all sites in the world and most of these sites have significantly smaller number of nodes. Another example could be airline networks.

In their model, Barabási and Albert[1] suggest that the underlying physical mechanisms responsible for generating a scale-free behavior are **growth** and **preferential attachment**. The former is connected to the addition of new nodes during the system's evolution, while the latter states that the nodes exhibit an intrinsic preference to link with a node which already has relatively more edges, compared to others. These mechanisms accomplish to lead to the formation of large clusters, the typical structure of scale-free networks, and yield a power-law degree distribution, but only when they are introduced simultaneously. Hence, these principles define the dynamics of such networks. Their combination may be summarized by the motto that "*the rich become richer*", indicating the uneven growth of edges

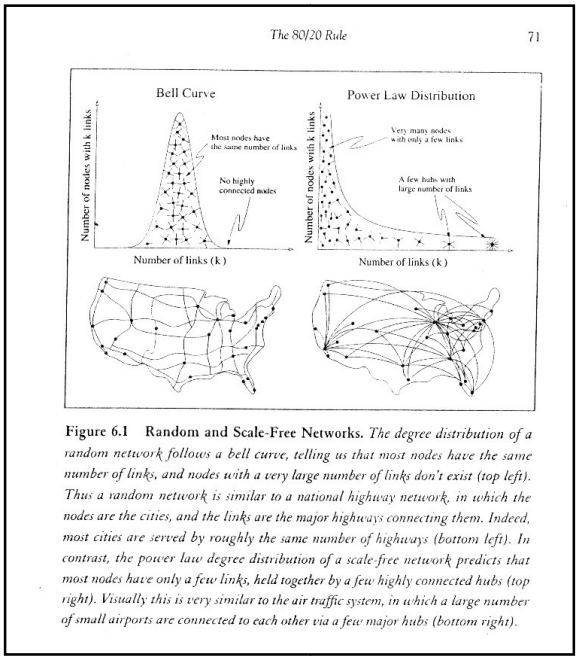



**Figure 6.1 Random and Scale-Free Networks.** *The degree distribution of a random network follows a bell curve, telling us that most nodes have the same number of links, and nodes with a very large number of links don't exist (top left). Thus a random network is similar to a national highway network, in which the nodes are the cities, and the links are the major highways connecting them. Indeed, most cities are served by roughly the same number of highways (bottom left). In contrast, the power law degree distribution of a scale-free network predicts that most nodes have only a few links, held together by a few highly connected hubs (top right). Visually this is very similar to the air traffic system, in which a large number of small airports are connected to each other via a few major hubs (bottom right).*

[1] Albert R. and Barabási A.-L., Statistical mechanics of complex networks, Rev. Mod. Phys. 74, 47–97 (2002). Barabàsi, *Linked: The New Science of Networks*, Perseus, Cambridge, MA, 2002.

from node to node.

The basic algorithm of the model is as follows:

STEP 1: The network begins with an initial network of $m_0$ nodes. It should be noted that $m_0 \geq 2$ and the degree of each node in the initial network should be at least 1, otherwise it will always remain disconnected from the rest of the network. Now nodes are added to this network one at a time. Each new node is connected to *m* pre-existing nodes and the probability of $\Pi(k_i) = \frac{k_i}{\sum_j k_j}$ connecting to the $i^{th}$ node is given by

STEP 2: *Growth* – For each given period of time we add a new node to the network.

STEP 3: Preferential Attachment – we assume that each new node connects to the existing network with two links. The probability that it will choose a given node is proportional to the number of links the chosen node has.

Therefore each time we add run *Step 2* and *Step 3* we add a node to the network, and the network grows:

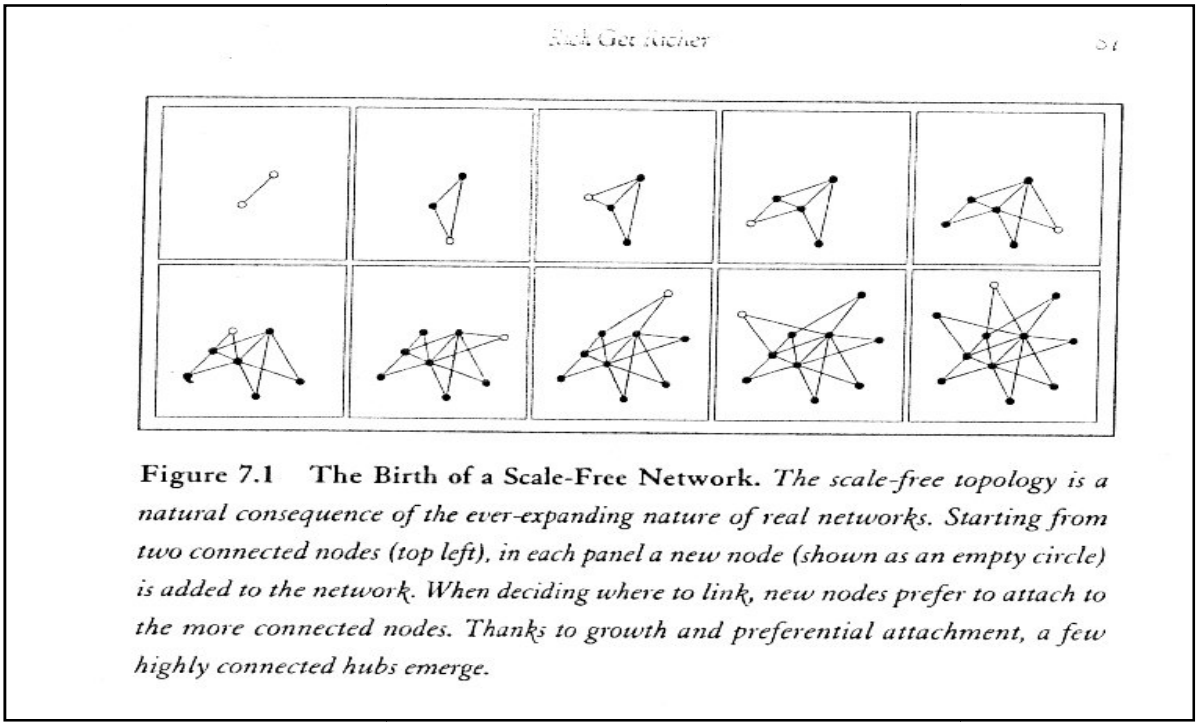
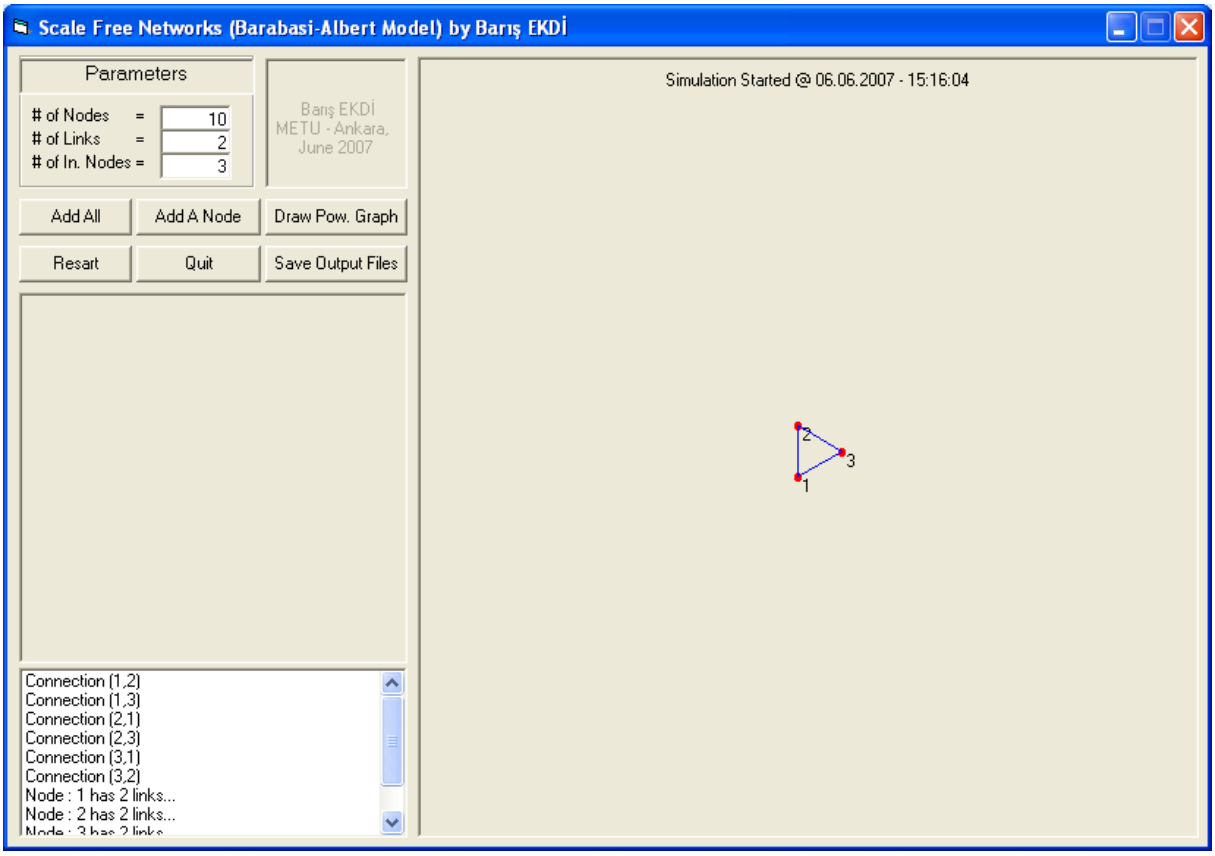


**Figure 7.1 The Birth of a Scale-Free Network.** *The scale-free topology is a natural consequence of the ever-expanding nature of real networks. Starting from two connected nodes (top left), in each panel a new node (shown as an empty circle) is added to the network. When deciding where to link, new nodes prefer to attach to the more connected nodes. Thanks to growth and preferential attachment, a few highly connected hubs emerge.*

# 2. THE PROGRAM

## 2.1. Main Features of the Program

The Program is written using Microsoft Visual Basic 6.0. The startup screenshot of the program is as follows:




*Parameters Frame:* A graphical user interface (GUI) is used for getting the required parameters from the user, in order to run the simulation. However, the user can also use the default parameters (Number of Nodes –to be added =10; Number of Nodes to be added per node=2; Number of Initial Nodes =3) loaded while starting the program. As the program started, a graph is drawn according to those values, and re-drawn whenever the user changes the number of nodes.

*Network Diagram Window:* Whenever a new node is added to the network, the node and its connections are shown in this window.

*Power Graph Window:* After the nodes are added, the relationship between the number of nodes and the number of links (ie. nodes with n links) is shown in *power graph window*.

*Detailed Process and Output Window:* This window is used in order to inform the user about almost each step of the process the program is executing. Therefore the user can check the calculations and save it for further use.

- There are six buttons provided to control the program:

*Add All:* Gets the parameters and starts the simulation. All of the nodes stated in the number of nodes box (minus the ones initially placed) are added one by one.

*Add A Node:* Just ONE node is added to the network at each click.

*Draw Pow. Graph:* Draws the power graph onto the *Power Graph Window.*

*Quit:* Ends the program end quits.

*Save Output Files:* Simulation output is saved to hard drive "C:\" in three files, with date and time stamp:

*Power Graph* is saved as a bitmap (.BMP) image file.

*Network Graph* is saved as a bitmap (.BMP) image file.

*Detailed Process* is saved as a text (.TXT) file.

### 2.2. How It Works? Pseudo-Algorithm of the Program

A simple pseudo-algorithm is given below in order to explain how the program runs:

1. Load default parameters and draw default graph with three nodes when the program is started – using the sub {*Initalize*}
2. Wait for user input. If the user;
   2.1. Hits <*Add All*> button go to sub {*Add_Nodes*} in order to add "n" nodes stated in the relevant box.
   2.2. Hits <*Add A Node*> button go to sub {*Add_Nodes*} but add only one node.
   2.3. Hits < *Draw Pow. Graph* > button go to subs {*Calculate_Edge3*}, {*Draw_PowGraph2*},{*Place_PowPoints*} in order to draw the power law graph.
   2.4. Hits <*Restart*> button, clear the windows and graph; go to sub {*Initalize*} for getting the default values and redraw and fill the windows.
   2.5. Hits <*Quit*> button END the program.
   2.6. Hits <*Save Output Files*> button go to sub { *Save_Output* }and save the files.

## 2.3. Main subprogram:

**{Add_Nodes}**

1.1. Get / check user defined parameters along with the button pressed. If it is <*Add All*> repeat the following process until the number of nodes (*NofNodes*) is reached, else if <*Add A Node*> button is clicked; do it only once.

1.2. For i=1 to *NofNodes* (For each Node);

1.2.1. For the first link;

Select two different random nodes;

Compare their probability of attachment (number of their edges/total edges);

Link the new node to the one which has higher probability;

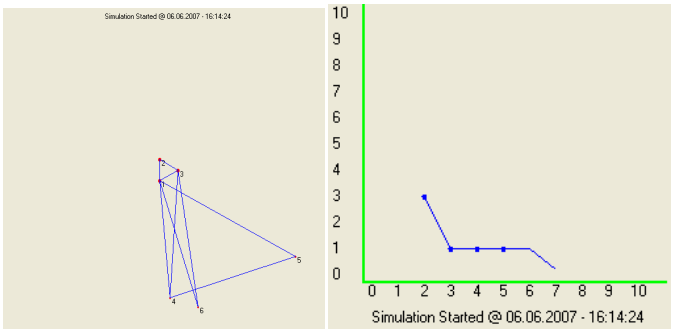
Draw the diagram accordingly.

1.2.2. Repeat process (1.2.1) above for the second link.

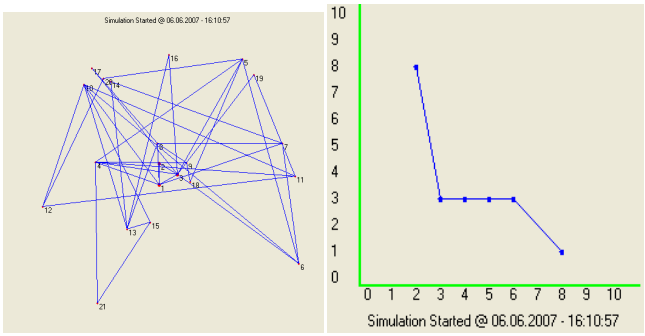
1.3. Add another node –if required – by repeating process (1.2.)

## 3. THE OUTPUT

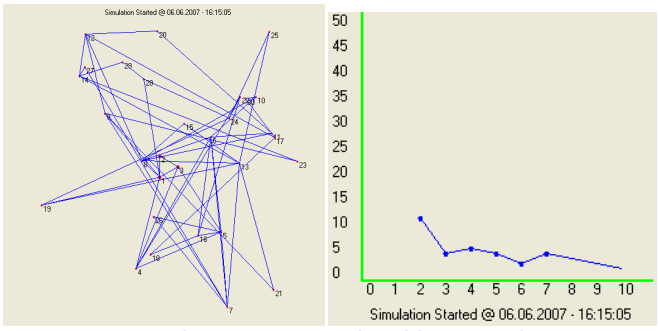
The results regarding eight simulations with different number of nodes {6, 20, 30, 100, 300, 1000, 2000, 10000 –respectively} is stated below. On the graphs at right hand side, Y axis represents the number of nodes with (k) links, and X axis represents the links (k). It is seen that, as the number of nodes grows, power law gains weight. (Detailed process is recorded to the relevant SFXXX_Process.TXT file for each of the simulation).



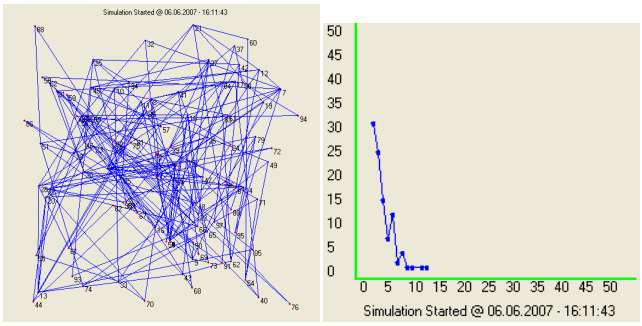

Scale Free Network With 6 Nodes



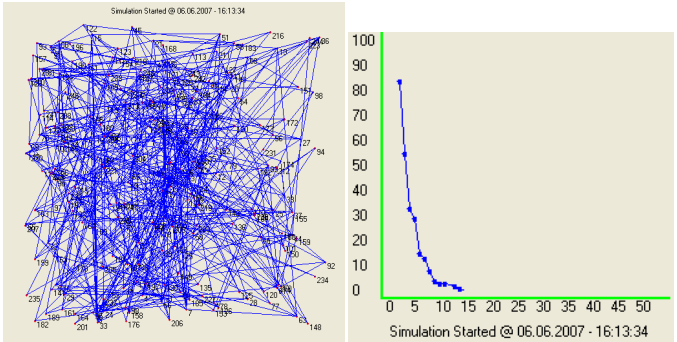

Scale Free Network With 20 Nodes



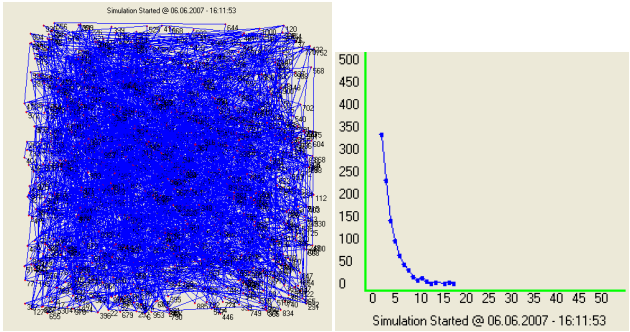

Scale Free Network With 30 Nodes



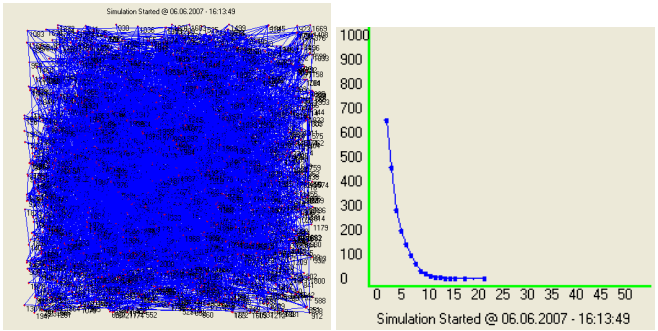

Scale Free Network With 100 Nodes



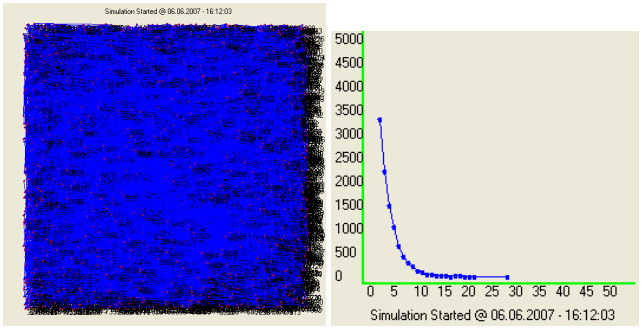

Scale Free Network With 300 Nodes




Scale Free Network With 1000 Nodes




Scale Free Network With 2000 Nodes




Scale Free Network With 10.000 Nodes

**-o0 THE END 0o-**