Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TU Delft
Delft University of Technology

.

Robustness optimization via link additions

Mari Carmen Sánchez Martínez
Student number 1558579

DIRECTOR: **Piet Van Mieghem- Fernando Kuipers**

SUPERVISOR: **Huijuan Wang – Javier Martín**

DATE: **13-07-2009**

TITLE: Robustness optimization via link additions

AUTHOR: Mari Carmen Sánchez Martínez

DIRECTOR: Piet Van Mieghem

SUPERVISOR: Huijuan Wang – Javier Martín

DATE: 13-07-2009

Overview

Robustness is the ability of networks to avoid malfunction. Networks could be subject to failures, viruses, removals of links and other attacks.

There are several researches about how to measure or improve the network robustness in large networks by doing small modifications such as adding new links or nodes. A small network modification is required because setting up links in real-world networks is expensive.

Different metrics can be used in order to measure robustness. This project focuses on the algebraic connectivity as the way of measuring connectivity. The main statement is that the larger the algebraic connectivity is, the more difficult to disconnect the network is.

The goal of this project is to design strategies to add a number (or a certain percentage) of links to a network such that the algebraic connectivity can be increased the most. Insights may come from results in mathematics, or the strategies for adding one link. In summary, the idea consists of optimizing a network in terms of algebraic connectivity by means of link additions.

Finally, different strategies will be evaluated on different types of networks such as the Barabasi-Albert power law graphs or Erdös-Rényi random graphs.

ÍNDEX

ÍNDEX OF FIGURES, TABLES AND EQUATIONS

INTRODUCTION

Complex networks describe a wide range of systems, with nontrivial topological features, in nature and society. These networks have patterns of connections between their elements that are neither pure regular nor pure random.

There are different ways of measuring robustness of networks and this project focuses on the increase of the algebraic connectivity of a graph. Algebraic connectivity corresponds to the second smallest eigenvalue of the Laplacian matrix. In order to calculate it, it is important to know the topology or the structure information of complex networks. This information will be obtained from the associated adjacency, Laplacian and other matrixes that describe the topology of the network.

Different types of complex networks are considered. Traditionally, the model of Erdös-Rényi random graph was used. Other models, such as the small-world and the scale-free graphs, are growing due to the small world and power law degree distributions observed in empirical networks.

The main goal of this project is to evaluate which is the best way to add a certain number of links in a network so that the algebraic connectivity *a(G)* can be increased the most. Furthermore, the influence of the number of links added to the network on the overall improvement has been studied.

The report is organized as follows. Chapter 1 introduces the concept of network robustness and the background about algebraic connectivity. Chapter 2 describes the complex network models that have been implemented and their features. Next chapters are focused on the simulation. Whereas chapter 3 tries to make a background about the simulations, chapter 4 defines the simulations and chapter 5 encloses the results of all the simulations. Finally, Chapter 6 summarizes the conclusions of this work and chapter 7 takes into account the future work that can be done in this research area.

CHAPTER 1. NETWORK ROBUSTNESS

Robustness is more than a property that defines a connected network; robustness is the key aspect in the design of computer networks. We can think about this term by taking into account the number of node or link failures that a network can accommodate before it becomes disconnected.

This chapter contains background information with the aim of providing a general view about networks, robustness and the concept of algebraic connectivity. It is also an introduction to understand the simulations and the goal of the project.

1.1. Complex Networks

Complex networks describe a wide range of natural and man-made systems. There are different types of networks such as WWW, Internet, social networks of acquaintances, paper citations or neural networks. A proper knowledge of their topology is required to thoroughly understand and predict the overall system performance.

A network is a set of items, which we will call nodes, with connections between them, called links [1]. Each node has a degree that is the number of links connected to it. Two nodes that are connected have an adjacency relation between them.

The representation of this network as a mathematical structure, in order to model relations between objects, is called Graph. We can say that a graph is a system taking the form of networks.

It is important to quantify graph’s robustness (or network robustness) and there are different ways to measure it.

- Node connectivity: It is the minimum number of nodes whose removal would disconnect the network.
- Link connectivity: It is the minimum number of links whose removal would disconnect the network.
- Algebraic connectivity: It is a spectral property of a graph, which is an important parameter in the analysis of various robustness-related problems. We have selected algebraic connectivity as the way to measure robustness. For this reason, it will be explained in detail in following sections [2].

To sum up, Figure 1 shows an example of a common graph. It represents a network with 8 nodes and 13 links. From this figure we can draw the following information: the node connectivity is 1 due to the fact that if node D is removed, the overall network will be disconnected. On the other hand, the link connectivity is equal to 2 because if the links that connect node D with B and C are removed, disconnection will appear again. Minimum degree nodes are A and H with a degree equal to 2 and the maximum degree node is D with degree equal to 5.

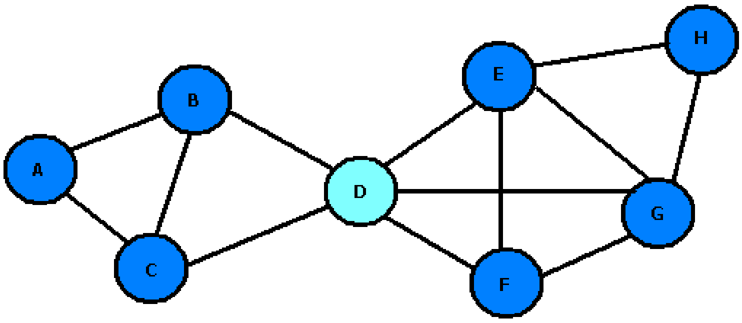


Figure 1. Network with N=8 nodes and L=13 links

1.2. Algebraic connectivity

Algebraic connectivity measures the extent to which it is difficult to divide the network into separate components. So, it measures the robustness with respect to the connectivity of a network.

Algebraic connectivity corresponds to the second smallest eigenvalue (that determines the scaling of an eigenvector) of the Laplacian matrix of a graph *G* (Equation 1.1), and it is only 0 if *G* is disconnected [2].

$$a(G) = \mu_{N-1} \quad (1.1)$$

It is known that algebraic connectivity is a lower bound on both the node and the link connectivity (1.2).

$$\mu_{N-1} \leq k_N \leq k_L \quad (1.2)$$

The Laplacian matrix *Q* of a graph *G* with *N* nodes is a N x N matrix and the expression is (1.3).

$$Q = \Delta - A \quad (1.3)$$

Where, Δ=diag(D_i) is the diagonal matrix with the degree D_i of each node (number of incident links on that node).
A is the adjacency matrix. It is a $N \times N$ matrix where the non-diagonal entry a_{ij} is the number of edges from vertex *i* to vertex *j*, and the diagonal entry a_{ii} is the number of edges from vertex *i* to itself.

The Laplacian eigenvalues of *Q* (the spectrum) are all real and non negative (1.4). They are always enumerated in non-decreasing order and repeated according to their multiplicity [3].

$$U_N \leq U_{N-1} \leq .... \leq U_1 \tag{1.4}$$

The Laplacian matrix of a graph and its eigenvalues can be used in several areas of mathematical research.

It is also important to take into consideration some theorems about algebraic connectivity such as the following ones:

- If *G* is a graph with *N* nodes and the graph *G+e* is obtained by adding a link between two nodes, the algebraic connectivity follows (1.5) [6].

$$0 \leq \mu_{N-1}(G+e) - \mu_{N-1}(G) \leq 2 \tag{1.5}$$

- The largest possible increase in algebraic connectivity occurs when all the nodes are connected between them except one link [7].

$$\mu_{N-1}(G+e) = \mu_{N-1}(G) + 2 \text{ where } G = K_N \setminus \{i, j\} \tag{1.6}$$

- If a graph *G* has a pendant node (node with degree 1) the algebraic connectivity is less than 1 or equal to 1. If the pendant node is not adjacent to the highest degree node the *a(G)* has to be strictly less than 1 [8].

Adjacency and degree matrixes have been introduced in the previous formula. The adjacency matrix *A* is a NxN matrix, where *N* is the number of nodes in the graph. If there is a link that connects node *i* with node *j* then the elements A(i,j) and A(j,i) are 1 (or in general the number of *i-j* links), otherwise they are 0. In computing processes, this matrix makes easy to find sub graphs and to reverse a directed graph.

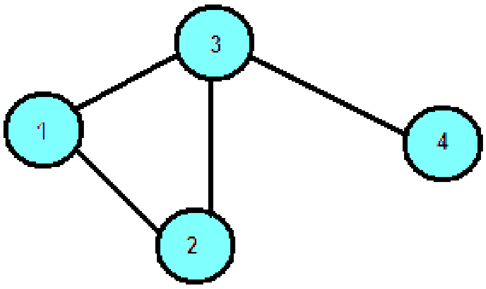
Concerning the incidence matrix *B*, it is a NxL matrix where B (i,j) and B(j,i) contains the link's endpoint data. The incidence list has the links represented by an array containing pairs of vertices (that the link connects)

Hereby, there is an example about the previous mentioned terminology and values. Figure 2 represents a network with 4 nodes and 4 links between these nodes. The adjacency matrix has an element equal to 1 between the nodes that are connected and a 0 between the others. The value is 1 because there is only one link between them but the value corresponds to the number of links that exist between a pair of nodes. On the other hand, the diagonal (Δ) has the degrees of the four nodes. After that we can obtain the Laplacian matrix and the second smallest eigenvalue which is the algebraic connectivity.

$$Q = \Delta - A_{NxN}$$

$$\Delta = diag(d_1, ..., d_N)$$

$$A_{NxN} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



$$Q_{NxN} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\downarrow$$

$$\mu_1, \mu_2, ..., \mu_N$$

Figure 2. Obtaining of the second smallest eigenvalue from the Laplacian matrix

To sum up, several properties about the algebraic connectivity are outlined below:

- The larger the algebraic connectivity is, the more difficult to cut a graph into disconnected components is.
- According to equation 1.2 a network with a relatively high algebraic connectivity is necessarily robust to both node and link failures.
- Its upper bound (in terms of node and link connectivity) provides worst case robustness to node and link failures.
- A graph is disconnected if and only if the algebraic connectivity is 0.
- The multiplicity of zero as an eigenvalue of the Laplacian matrix *Q* is equal to the number of disconnected components of the graph *G*.

Before finishing, it is also important to introduce the concept of algebraic distance because one of the strategies for link addition is based on it.

The algebraic distance is a linear distance metric commonly used in computer vision applications because of its simple form and standard matrix based on least mean square estimation operations.

In geometry, a distance function on a given set *M* is a function d: $M \times M \rightarrow R$, where *R* denotes the set of real numbers, that satisfies some conditions that are described below.

- Distance is positive between two different points (1.7)

$$d(x, y) \geq 0 \tag{1.7}$$

- Distances is zero from a point to itself (1.8)

$$d(x, x) = d(y, y) = 0 \tag{1.8}$$

- The distance between points x and y is the same in either direction (1.9)

$$d(x, y) = d(y, x) \tag{1.9}$$

- The distance between two points is the shortest distance along any path (triangle inequality) (1.10)

$$d(x, z) \leq d(x, y) + d(y, z) \tag{1.10}$$

The usual definition of distance between two real numbers *x* and *y* is: d(*x*,*y*) = |*x* − *y*|. This definition satisfies the three conditions above, and corresponds to the standard topology of the real line.

.

CHAPTER 2. COMPLEX NETWORK MODELS

To model the topology of computer networks, two models have been taken into account: the random graph of Erdös-Rènyi and the scale-free graph of Barabasi-Albert [1].

This section is focused on the models' description and properties.

2.1. Models

Power law degree distributions have been observed in different complex real-world networks. The first growing network model is Barabasi-Albert model. There are networks that do not grow in a pure way because they suffer link or node removals, link rewiring or other perturbations. For this reason, we are going to analyze also the Erdös-Rényi random graph that is the simplest model to describe a complex network.

2.1.1. Erdös- Rényi (ER) random graph

This model was proposed by Erdös-Rényi [5]. One of its main features is that it is the simplest model to describe a complex network. Furthermore, it has a homogeneous structure of a random graph that implies an invariant robustness under random node and link failures.

In a random graph $G_p(N)$, N is the number of nodes in a graph and p is the probability of having a link between two nodes. Undirected links are placed at a random between a fixed numbers of N nodes to create a network. Each of the $\frac{n}{2}*(n-1)$ possible links is independently present with probability p and the number of links connected to each node is distributed according to a binomial distribution. A random graph is connected if and only if $p \geq \frac{\log N}{N} = p_c$ when N is large. This value p_c is called the critical link density; the graph will be disconnected if p is below this result [9]. For this reason, $p = \alpha p_c$ where α is a free election parameter. We can also deduce that the probability of a random graph being connected is approximately the probability that no node has the degree equal to 0 for large N [4].

This model is reasonably accurate for peer to peer and ad-hoc networks.

Besides that, the growing observation that real-world networks do not follow the prediction of random graph has generated the appearance of other types of networks.

2.1.2. The Barabasi-Albert (BA) power law graph

Barabasi-Albert is a model for evolving networks. This model is a scale-free graph that implies a non trivial robustness to random node and link failures. The scale free networks are the ones whose degree distribution (2.1) follows a power law. The power law implies that the degree distribution of these networks has no characteristic scale.

$$P(k) \approx k^{(-r)} \qquad (2.1)$$

Where, r is the degree exponent.

Barabasi [10] introduced the concept of growth and preferential attachment of nodes. Nodes with larger degree are more likely candidates to be attached by new nodes. To sum up, the probability of being connected is proportional to the degree d_i of the node. The resulting graph possesses a power-law degree distribution. Moreover, links are undirected, so there is no distinction between in and out degrees.

The model starts with a number m_0 of fully-meshed nodes. It is important to mention that $m_0 \geq 2$. New nodes are added to the network one by one. Each node is connected to m existing nodes in the network. An existing node is chosen to be connected to the new node with a probability proportional to the degree of this node. Hence, new nodes are preferentially linked with the most connected nodes of the network. The probability p_i that a new node is linked with *i* is (2.2):

$$p_i = \frac{k_i}{\sum_j k_j} \qquad (2.2)$$

Where, k_i is the degree of the node i.

We are in front of an inhomogeneous network where most nodes have very few connections and only a small number of nodes have many connections. This feature makes this type of networks error tolerant but really vulnerable to attacks. Error tolerant means that they have a robust connectivity against random failures but the problem appears if the most connected node is removed, attacked or if it fails, then we lose the connection between a high number of nodes.

Power law degree distribution is followed by many natural and artificial networks such as scientific collaboration or World Wide Web.

2.1.3. Comparison between models

The network robustness subject to link failures differ between the ER random graphs and BA power law graphs. Hence, it is important to consider both models and to make a comparison between them.

Figure 4 shows the difference between BA power law network and ER random graph. Y axis represents the number of nodes with k links and x axis the number of links *k*. In the random graph, most nodes have the same number of links and there are no highly connected nodes. Furthermore, a small number of nodes have few or no links. The tail (high *k* region) of the degree distribution *P(k)* decreases exponentially, which indicates that nodes that significantly deviate from the average are extremely rare. On the other side, power law degree distribution shows that there are many nodes with only a few links and few nodes with a large number of links. We can call these nodes hubs. This situation is generated by the preferential attachment.

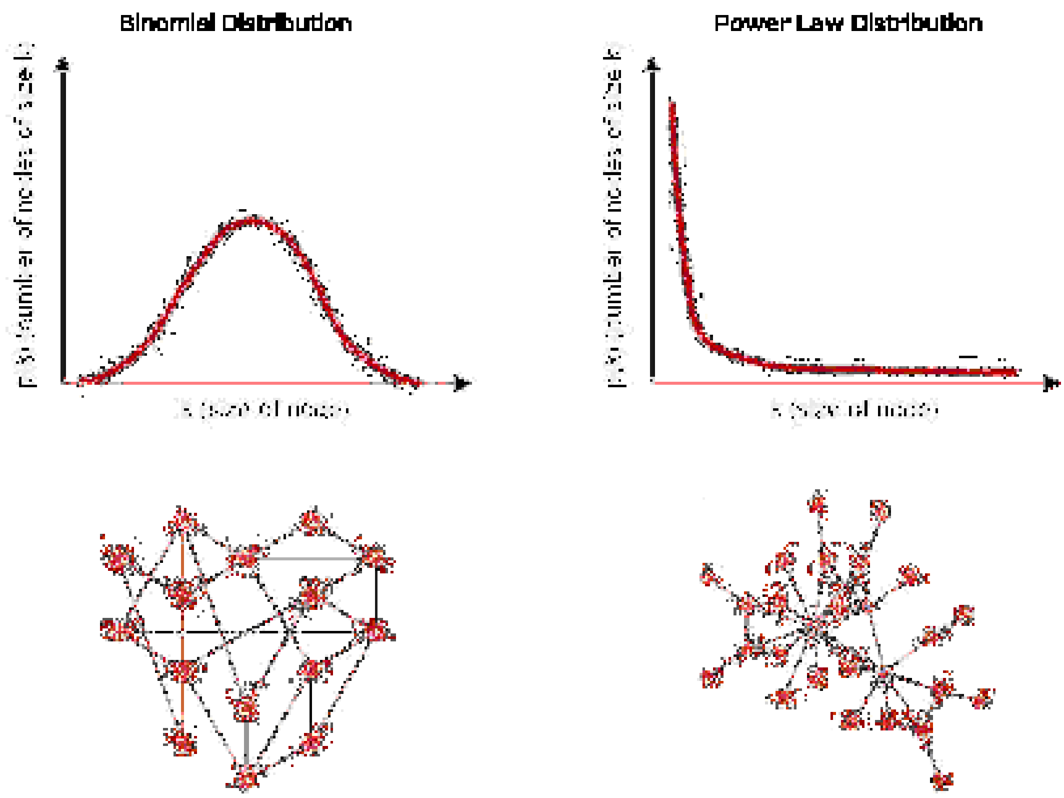


Figure 3. Comparison of degree distribution between ER random graph and power law network

CHAPTER 3. BACKGROUND INFORMATION

The aim of the project is to optimize the robustness of the network by adding a set of links between different pair of nodes that are not still connected. Topological features of the node pairs have to be considered.

The aim of this section is to provide a background about other researches and studies concerning the same topic and to define the software that is going to be used in the simulations.

3.1 Link addition strategies

A series of strategies for adding links have been established. The goal is to analyze the behaviour of the network in terms of robustness, which is characterized by the algebraic connectivity, when a certain number of links are added. Specifically, we evaluate the influence of the number of added links and the strategies applied on the algebraic connectivity.

Being *N* the number of nodes and *L* the number of links, the number of possibilities of adding a link between two non-connected nodes in a network G(N,L) is $\binom{N}{2} - L$. The node pair *{i,j}* where we add a link can be selected based on some properties of the node pair such as the node degree and the distance between the two nodes.

As we have mentioned before, two models have been taken into account, the Erdös-Rényi and the Barabasi-Albert model. The research is carried out independently but in a comparative way over these two models.

We propose the following strategies to add a number of links such that the algebraic connectivity can be increased as much as possible. Some of them are based on topological metrics of nodes, while others on the eigenvector properties of the network.

- Addition of links in a random way.
- Addition of links between the minimum degree node to a random no connected node.
- Addition of links taking into account algebraic distance (eigenvector) so that to add a link between a node pair with the maximum $|u_i - u_j|$, the absolute difference between the i-th and j-th elements of the Fiedler vector *u* of *G*. As we have explained before, the Fiedler vector u is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix of the graph. The set of links can be added one by one recursively or they can be added at once to the set of nodes pairs with the maximum eigenvector distance $|u_i - u_j|$.

- Addition of a certain number of links between pairs of nodes with the minimum degree.
- Addition of links between the maximum degree nodes to random non-connected ones.

There are two options for link additions. The set of links can be added once based only on the original graph. Otherwise, the set of links can be added one by one based on the updated network structure and calculate the second smallest eigenvalue each time a link is added to the network in order to do a precise calculation step by step and to add a link one by one instead of all together. The drawback of the latter is the high amount of needed resources. As we will see in future chapter there are strategies where the result is equal in both types of links addition.

To implement the simulations a set of initial parameters have been defined, such as the number of nodes or the link density of the node. These parameters will be explained in next section.

The number of links that are going to be added is variable between 10 and 50 depending on the simulation.

If a link *e* is added to the network *G* the Laplacian spectrum (set of all N Laplacian eigenvalues) corresponds to the Equation (3.1)

$$0 = \mu_N(G) = \mu_N(G+e) \le \mu_{N-1}(G) \le \mu_{N-1}(G+e) \le \ldots \le \mu_1(G) \le \mu_1(G+e) \qquad (3.1)$$

3.2 Previous research

It is well known that graphs with large *a(G)* have many applications. Extensive simulations show that the algebraic connectivity increases with the increasing node and link connectivity, implying that the algebraic connectivity is a good measure of robustness in complex networks [9]. On the other hand there is a difficulty of applying strategies to optimize the algebraic connectivity by adding a link based on topological characteristic. This is due to the fact that a strategy that works for a given type of graph may not work in other types of networks.

Another feature proved by Juhaz [11] is the asymptotic behaviour of the algebraic connectivity in Erdös-Rényi random graph *Gp(N).*

Previous researches about the best way of adding one link are in literature [12]. Insights can be obtained from the optimization of the algebraic connectivity via one link addition. For example, the addition of 1 link between a pair of nodes with the maximum algebraic distance performs generally better than adding links between nodes that have been randomly chosen. It happens in both models (ER and BA).

Recall that the algebraic connectivity of a graph is a measure of robustness regarding to connectivity.

3.3 Used software

Different software has been used in order to program the strategies, to analyze the obtained data and so on.

Simulations have been obtained by creating a C code that computes the second and third smallest eigenvalues of a network depending on different parameters and models. It allows the implementation of the six different strategies and it fits all the requirements of the project. Chapter 4 explains in detail the features of the program. The code has been created with PSPad editor.

PSPad editor is a free editor of Microsoft Windows that allows the use of different programming environments and it has syntax highlighting. It is useful to use this software in order to create the strategies and store the data. It is easier to modify the code with this program than by console using pico or vi editors.

When the code is ready it is uploaded to TUDelft cluster where the simulations are carried out. For this action WinSCP is used. TUDelft cluster has all the needed libraries and files required for the proper work of the program. The clusters' name is asterix.et.tudelft.nl and idefix.et.tudelt.nl, the latter was used in substitution of the first one due to technical problems.

Once the simulation finished a file is generated. The output file is exported and analyzed with Igor Pro software.

IGOR PRO is registered software of Wavemetrics. It is a powerful and extensible scientific graphing, data analysis, programming software tool, etc. Moreover, there is a wide range of manuals and examples explaining how to use it and it is customizable depending on the requirements of the project.

A graph can be created by means of the definition of the waves. The term "wave" is used to describer an Igor object that contains an array of numbers. The purpose is to store, analyze, transform and display waves. A histogram is a type of graph that shows the variability of a process (its distribution). It consists of a number of bins of equal width, starting at a particular X value. The number of bins is determined by the number of points in the destination wave. After the histogram, each data value of the destination wave contains a count of the number of data values in the source wave which fall in a given bin. Finally, this information is used for obtaining the results.

CHAPTER 4. SIMULATIONS DEFINITION

Some parameters have to be settled before doing the simulations. Moreover, it is important to have a clear view about the simulation process and the features of the strategies. This chapter tries to clarify all these concepts.

4.1 Parameters of simulations

The simulations are carried on fixed initial parameters in each of the studied models to analyze and to compare the obtained results.

Lets $G(N,L)$ be a simple graph on the node set N={1,…,N} of N nodes. We consider only undirected graphs. The degree of a node is denoted by d_i. For each combination of N and p we are going to generate 10^4 independent $G_p(N)$ graphs.

On the other hand, we have to set a link probability p, which represents the density of the ER random graph *Gp(N)* and the average degree *m* of the BA power law graphs.

The values of the initial parameters are detailed below.

1- Barabasi-Albert model

a. Number of nodes N =100
b. Link density $p = \frac{2m}{(N-1)}$ Average degree (m) = 3 (for large N)
c. Number of added links l ranges from 10 to 50.

2- Erdös-Rényi:

a. Number of nodes N=100
b. Link density $p = \frac{\log N}{N}\alpha$. Hence α is an arbitrary value that can be chosen. Link density is equal to 0.4 or 0.06 depending on the simulation.
c. Number of added links l ranges from 10 to 50.

The value of the link probability can also change in order to evaluate different types of networks. The selection of p value in Erdös Rényi is a critical point due to the fact that p has influence on the behaviour of the system. When the link density is smaller than the critical link density p_c there are few links and many small disconnected components. On the other hand, if we have a high density p that is larger than p_c an extensive fraction of nodes is joined together forming a single giant component.

Another previous consideration is that all the nodes and links are equal. It is an undirected network with only a single type of node and link.

4.2 Implementation

The results of the simulations come from a code that computes the second and third smallest eigenvalue of the Laplacian matrix depending on different parameters and models. The C program has been programmed in order to fit the requirements of the project and to implement the explained strategies.

Firstly, it is important to understand the program. The main function has the declaration of the variables, the definition of the original network and the functions where the strategies for link addition are programmed.

Three different files are involved in the simulations: usedlinks.c, inputfile and linkeigQ3way.ran (outputfile).

Usedlinks.c includes the code of all the strategies that are going to be implemented, inputfile includes the input parameters of the simulations (we call parameters values such as number of nodes, probability of link addition or number of added links). Finally, linkeigQ3way.ran is the output file that is created once the simulation has finished and where the obtained information is stored in order to be saved. An example of the output file appears in Figure 5. It includes the second and third smallest eigenvalue of the original network and the algebraic connectivity after implementing each strategy.

Iteration	$a(G_0)$	μ_3	$a(G_1)$	$a(G_2)$	$a(G_3)$
1	1.290305	1.542674	1.628558	1.698495	1.760493
2	0.946163	1.391401	1.849190	1.997484	2.012504
3	1.264425	1.407968	1.562888	1.618135	1.819350
4	1.369407	1.514684	1.785661	1.874557	1.903305
5	1.466389	1.492758	1.820102	1.853949	1.871667
6	1.257033	1.440381	1.754741	1.805083	1.845658
7	1.349457	1.390890	1.819687	1.893751	1.954225
8	1.374437	1.433676	1.752328	1.793985	1.796964
9	1.378839	1.477938	1.673537	1.809277	1.821798
10	1.347613	1.403327	1.725357	1.814721	1.941394

Figure 5. Output's file structure (10/10000 rows)

We are going to explain the main file usedlinks.c and especially how the strategies are programmed.

We have to pay attention to important functions which are the generation of graphs and the function to calculate the second smallest eigenvalue of the Laplacian matrix. Moreover, two models are taken into account, Erdös-Rényi (ER) and Barabasi-Albert (BA).

For computing all the eigenvalues of the Laplacian matrix we have used LAPACK implementation of the QR-algorithm. LAPACK is a Linear Algebra Package written in Fortran90 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems.

This section of usedlinks.c also contains the definition of all the parameters and libraries that are going to be used, the obtaining of the parameters that the user writes in the inputfile, the memory allocation and the generation of the original network and its matrixes.
In this first section, the calculation of the eigenvalues corresponds to the case of the original network, without any link addition.

Next step is to apply the strategies in order to add the links. Strategies are adaptable to the number of links that the user wants to add. Each strategy (function) returns the second smallest eigenvalue (algebraic connectivity) after the addition of links. This result is written in linkeigQ3way.ran updating the files with the results of each of the 10^4 independent graphs.

There is a common code in all the strategies which function is:
- Initialize the adjacency matrix after every link addition
- Change the value of the number of adjacent nodes and set the linkcount to one every link addition. The linkcount is the value that shows if a pair of nodes is still connected or not.

Finally, the code is executed using the Makefile and Putty software in TUDelft cluster Asterix / Idefix.

4.3 Strategies

The aim of this section is to explain explicitly the strategies that have been mentioned in chapter 2 to understand better how they work.

Three of the strategies are derived from previous research on adding one link [12]. A set of links are added. Another three are new ones. An example about where the links are going to be added is included on each description. The sample network has 5 nodes and 5 links between them.

4.3.1 Strategy 1: random addition

Strategy one is the simplest one and it adds links in a random way between nodes that are not connected yet. Information about the nodes (for example the degree of the node) is not needed to apply this strategy.

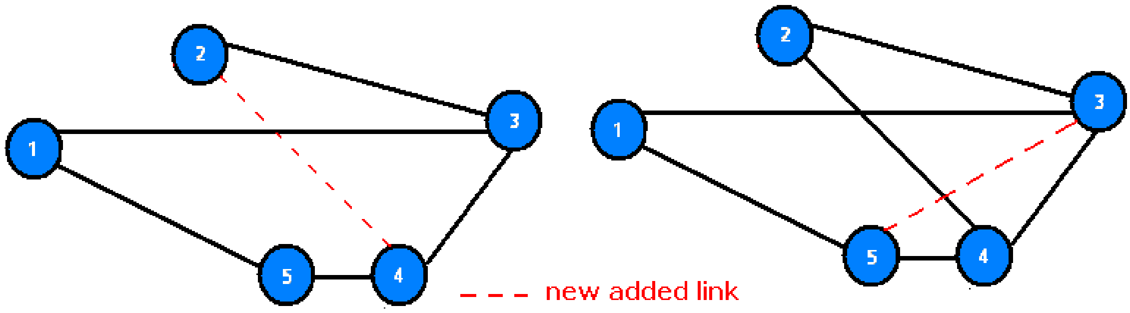


Figure 5. Addition of 2 links using strategy 1 (ER and BA model)

A link is added randomly between two nodes that are not connected. Given *l* links are going to get added into the network, the process of adding one link is repeated *l* times. The adjacency matrix is updated whenever a link is added. See Figure 5.

There is a checking function to know if there is already a link between both nodes or if we are selecting the same node for both ends of the link. After the *l* links have been added the eigenvalues (second smallest eigenvalue) are computed.

4.3.2 Strategy 2: minimum degree node

Strategy two involves adding links between the minimum degree node in the network and a random other one. The idea is to increase the connection of the lowest degree node with the rest of the network in order to avoid its possible low performance.

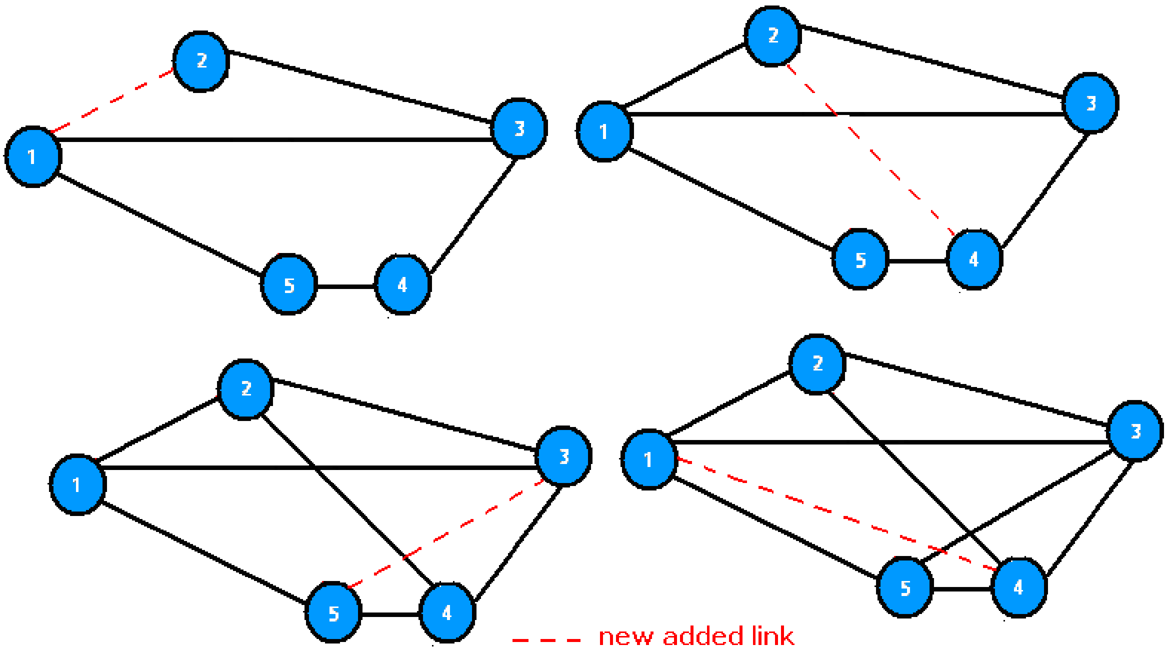


Figure 6. Addition of 4 links using strategy 2 (ER and BA model)

As you can see in Figure 6 the link is always added between the node with the lowest degree and another one selected randomly. If more than one nodes have the same lowest degree, one of them will be selected.

To implement this strategy we need local information, specifically, the degree of the nodes in the network.

The strategy finds which node has the lowest degree and adds a link between this node and another random one. The next step is, as in previous strategy, modify the adjacency and Laplacian matrix by including the link added. After that, the addition is done between the lowest degree node and a random node.

There is also a checking function in order to know if there is already a link between both nodes or if we are selecting the same node for both ends of the link. After the addition of the complete number of links the calculation of the eigenvalue (second smallest eigenvalue) is done and placed in the result file.

4.3.3 Strategy 3: eigenvector

Strategy 3 adds links based on the algebraic distance (eigenvector). A link is added between a node pair with the maximum $|u_i - u_j|$. It corresponds to the maximum absolute difference between the i-th and j-th elements of the Fiedler vector of G. This strategy is based on the fact that $\mu_{N-1}(G+e)$ tends to be large if $\alpha = |u_i - u_j|$ is large [12]. The drawback of this strategy is that we need to have the knowledge of the topology of the whole network.

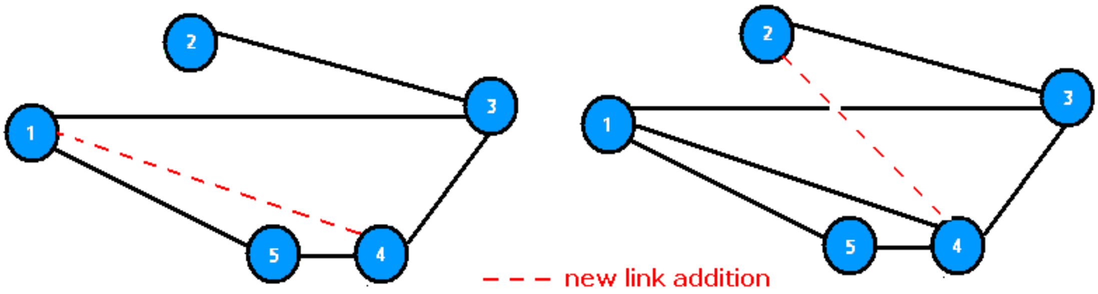


Figure 7. Addition of 2 links using strategy 3 (ER and BA model)

In order to implement this strategy we calculate the algebraic distance between all the combinations of pair of nodes and select the two non-connected nodes with the maximum algebraic distance. Before link addition, there is a checking function between already connected nodes. After adding a link, the adjacency and Laplacian matrix have to be updated. The selection continues until all the links are added between the selected nodes. Finally, the algebraic connectivity is computed after the set of l links have been added all together.

4.3.4 Strategy 4: Similar degree nodes

The goal of this strategy is to add a certain number of links between nodes with the minimum degree (when the number of added links is low) or between similar degree nodes in order to evaluate the increase in the algebraic connectivity. The idea is to check if adding links between lower degree nodes effectively improves the general network connectivity.

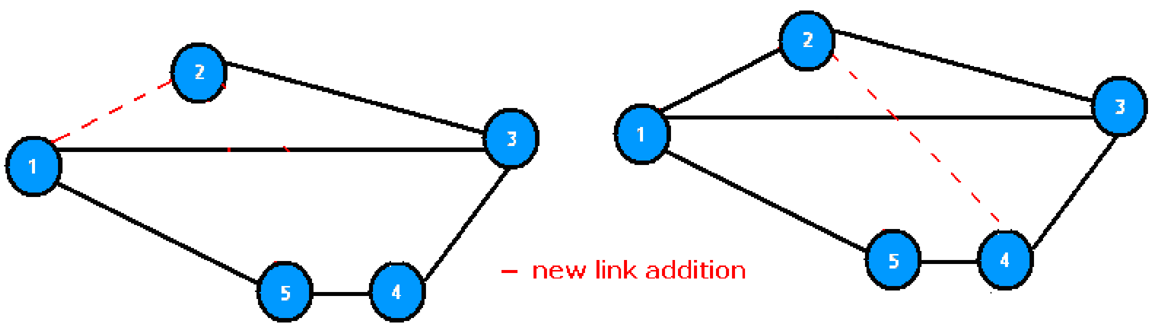


Figure 8. Addition of 2 link using strategy 4 (ER and BA model)

Firstly, the strategy searches the 20 nodes with the lowest degrees. The nodes are arranged from the lowest degree one till the highest degree node. Then we start the addition between the nodes following sequential way (not random) thanks to the previous arrangement. Sequential way means that, the first link addition is between the lowest degree node and the 2nd lowest degree node, the second link addition is between the 3rd and the 4th lowest degree nodes and so on. In case of having already one link between a pair of nodes the program selects randomly another one of the network. 20 nodes are chosen because we want to add ten links using all the nodes. In case of adding 10 or 50 links, 20 or 100 nodes will be chosen. The need of having the double of nodes than links we want to add is because the goal of the strategy is to add links between pairs of nodes with similar degree. We will have pairs of nodes with similar degree that are connected.

In Figure 8 we can see how the first time the links is added between node 2 and 1, because they are the ones with the minimum degree. The following addition is done between node 4 and 5 that are the next ones with lower degree but there is already a link between them so the addition is between node 4 and a random one.

After the link addition, as in previous strategies, the calculation of the second smallest eigenvalue is done and it is stored in the output file.

4.3.5 Strategy 5: maximum degree node

The fifth strategy is related with strategy 2, the goal is to add links between the maximum degree node (instead of the minimum degree node used in strategy 2) and a random other node.

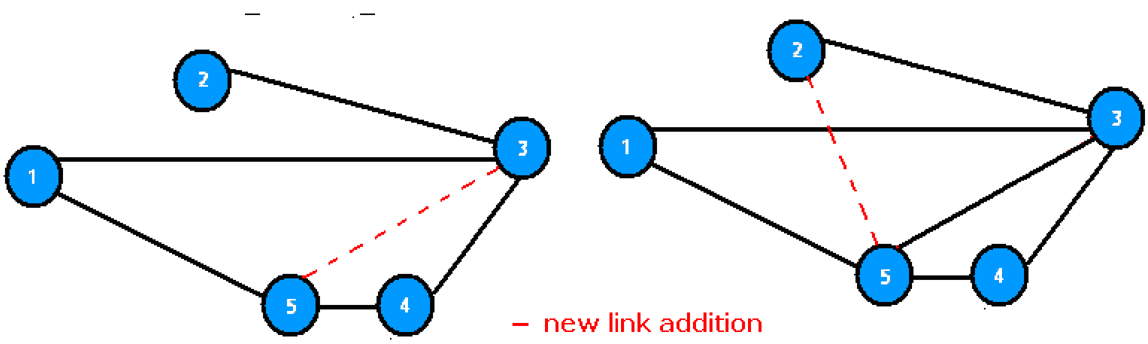


Figure 9. Strategy 5 implementation in Erdös- Rényi and in Barabasi-Albert

Moreover, we have also checked if there is a link already between them and if not a link is added.

This strategy would present a problem. After each link addition, the maximal degree node remains the same. It is possible that, after several link additions, the maximum degree node is connected with all the other network nodes. Hence, no links can be added any more. To prevent this situation a modification of the strategy is done. If the node is capable of supporting the number of links that the user wants to add, they will be added without any problem. On the other hand, if the maximum degree node can withstand any link they will be added to it. The others will be added to the second or next maximum degree node. Finally, if the maximum degree node is already fully connected with the other nodes, links will be added one by one between the second maximum degree node and a random other node.

4.3.6 Strategy 6: Link by link addition

Strategy 6 is based on the same concept as strategy 3. The link is added between nodes with a maximum algebraic distance. The difference lies in the fact that this strategy makes the adhesion link by link, each time a link is added the calculation of the algebraic connectivity is computed. On the other hand, using strategy 3 the addition is done once, all the links are added together and then the algebraic connectivity calculations are done.

In order to implement this strategy we calculate the algebraic distance between all the combinations of node pairs and select the two non-connected nodes with the maximum algebraic distance. Fiedler vector of the updated network has to be recomputed for the next addition of a link. In next link addition, the nodes will be other ones because there is a checking function between already connected nodes. Finally, the algebraic connectivity is computed after each link addition. The results are saved in the output file.

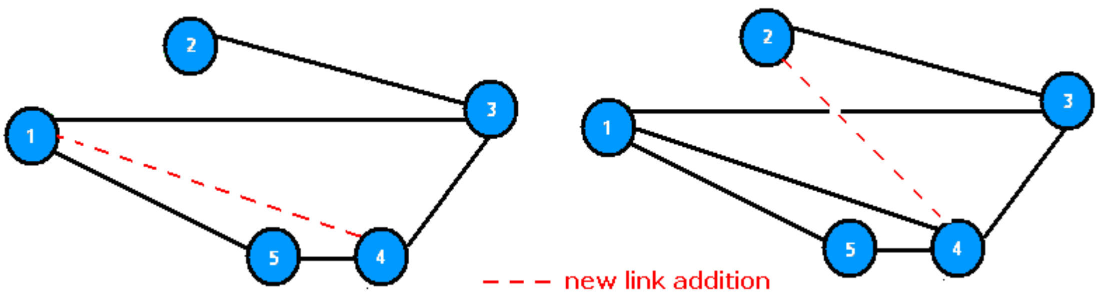


Figure 10. Addition of 2 links using strategy 6 (ER and BA model)

CHAPTER 5. SIMULATION RESULTS

In this section we present a comprehensive set of simulation results taking into account both complex network models: the scale –free graph of Barabasi-Albert and the random graph of Erdös-Rényi. The description of the models appears in chapter 2.

Once all the strategies were programmed, the simulations can be carried out. As we have previously mentioned they were carried out in the TUDelft Idefix cluster. All the used libraries, the inputfile and makefile are in the simulation's folder. The makefile is a text file that manages the compilation of the program. It is like a database that informs about the dependences between the different parts of the project. After compiling the code to check that there are not errors the following step is executed. Next screenshot shows which commands are used to compile and launch the program (Figure 5).

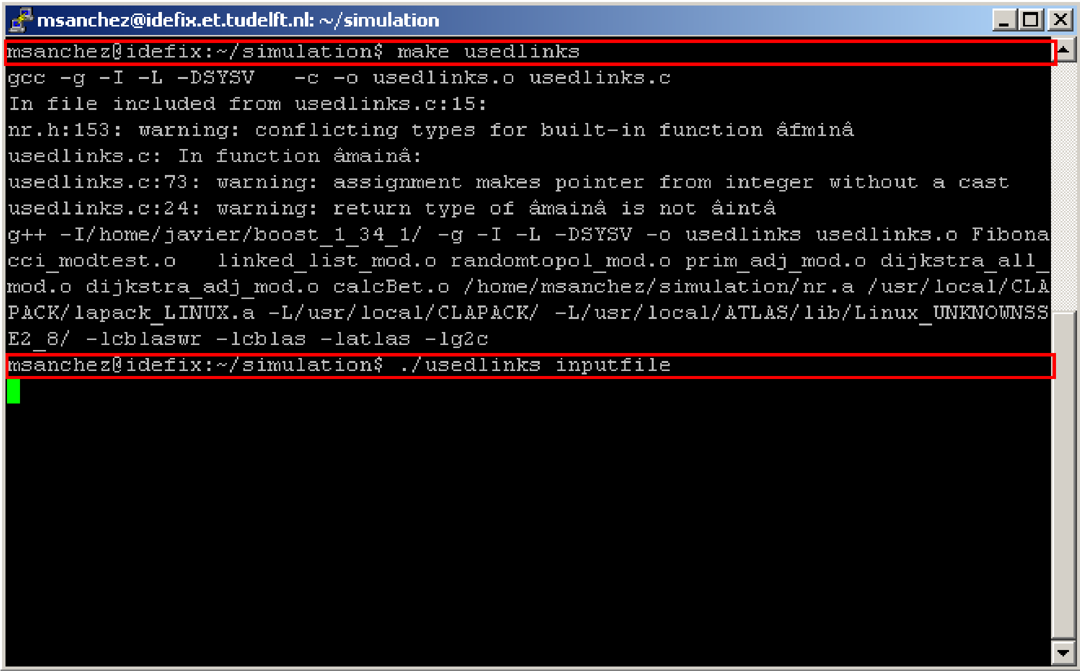


Figure 11. Commands to compile and launch the program usedlinks.c in the cluster Idefix.

When the simulation finishes the output file is generated. The processing of the data is the subsequent. Through WinSCP we pick the output file. Subsequently, it is uploaded in Igor pro. As you can see in the screenshot (Figure 11) the upload is done with the commands Data→ Load Wave → Load General Text. Then we have to assign a name for each wave and to create a table. With all these data the histogram is generated. To create it, we need to write a code that is placed in the procedure window (See Annex B).

After that, we only have to select the number of bins used to generate the histogram. In Figure 12 you can see the commands:

Make/N=128 seqX=p+1; where a sequence of 128 binds is created.
Funchistogram(src,xdstname,dstname,128,seqX); it allows deciding which wave is going to be plotted. Src corresponds to the source wave, xdstname and ydstname are the name for x and y axis of the destination wave, 128 is the number of binds and seqX is the sequence that you have previously created.

Finally, we will obtain the histogram where the waves with the evolution of algebraic connectivity are shown. The histogram represents the frequency of each value of algebraic connectivity in the network.

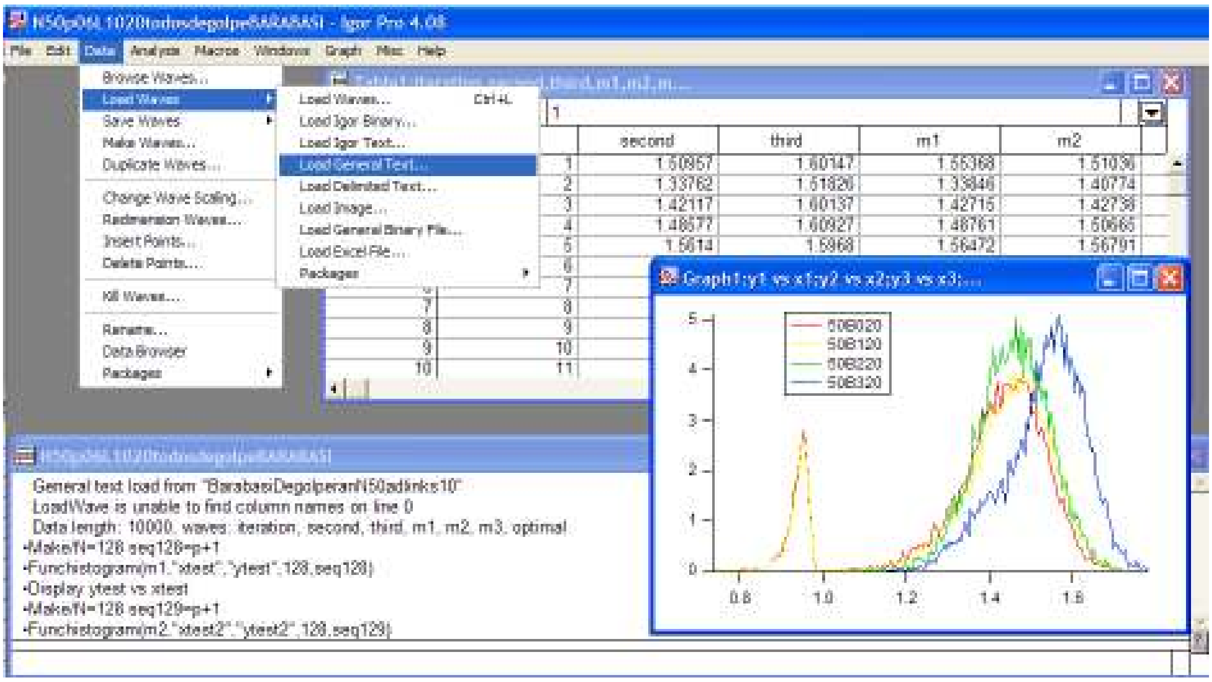


Figure 12. IGOR Pro interface and results.

Before having a look to the results we should briefly mention the meaning of the legend that is going to appear in the histograms. Each wave (the algebraic connectivity after applying each strategy) is represented with one specific colour and marker. About the legend, hereby (Equation 4.1) you can find an example to understand it.

$$a(G_x(L+l)) \quad (4.1)$$

$$E(a(G_x)) \quad (4.2)$$

$$I\_a(G_x(L+l)) \quad (4.3)$$

Where:
a(G) is the algebraic connectivity.
E(a(G) is the average of the algebraic connectivity.
I_a(G) is the increase in the algebraic connectivity with reference to the original network.

x is the implemented strategy (the numbers of the strategies are the same that have been listed above).
L represents the number of links of the original network.
l is the number of links added to the network.
Henceforward, the results of the simulations with 100 nodes are shown.

5.1 Barabasi-Albert Results

We have divided the results depending on the network model. We are going to start with Barabasi-Albert model. Two different analyses could be considered, first the algebraic connectivity is computed only after the set of links have completely been added and second we calculate the algebraic connectivity of each intermediate network and the set of links are added one by one. The first way of calculating the algebraic connectivity is implemented with strategies 1 to 5 because of the computations complexity of the second one. The second way is implemented only with strategy 6.

5.1.1 Results of strategies 1, 2 and 3

Graphs are the best way to get the first fast impressions and statements about the results of the simulations.

First of all we are going to analyze the results of Barabasi-Albert with N=100 and m=3. Figure 13 shows the improvement of the algebraic connectivity as the number of links increases.

Strategy 3 performs the best in all the simulations followed by strategy 2 and strategy 1. This fact means that the most efficient way for adding links in terms of optimizing the algebraic connectivity is to add links between nodes with the maximum algebraic distance. However, adding links randomly improves relatively less the *a(G)* of the network in comparison with the other two strategies. The addition of links between nodes with low degree can hamper the increase of *a(G)* but as it will be shown later (Section 5.5) adding a link between a high and a low degree node does not imply a higher increase in algebraic connectivity than implementing other strategies. Finally, we can also notice that as the number of links increases the performance of strategies 2 and 3 seems to get closer. On the other side, strategy 2 works well because we are forcing that one of the end points of an added link was the minimum degree node that is the one that has less connections.

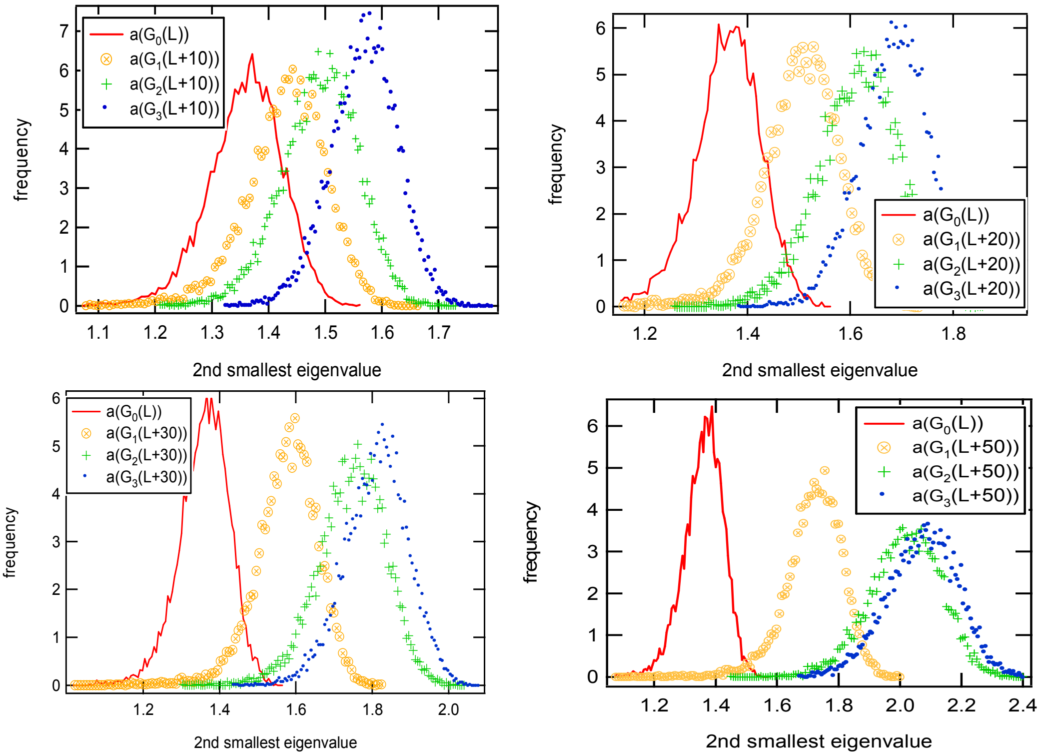


Figure 13. Algebraic connectivity in Barabasi-Albert model after adding 10,20,30 and 50 links with N=100 and m=3 using strategies 1,2 and 3.

5.1.2 Results of strategy 4

The next step is to evaluate strategy 4. It consists of adding a link between a certain numbers of minimum degree nodes. Four cases have been considered: from the addition of 10 links between 20 nodes with the smallest degrees until the addition of 50 links taking into account all the nodes of the network. The selection of the number of nodes has not followed any rule but the idea is to add a link between the node with the minimum degree and the next one following an increasing order. Therefore, the number of nodes to be selected doubles the number of links to be added. In conclusion, we are adding links between pairs of nodes with similar degree.

The results of the simulation show how strategy 3 continues being the one that performs better in front of the others. It is also visible that the improvement by strategy 4 is not high if the number of added links is low. This situation is due to the fact that adding a link between both nodes with the minimum degree can not generate a high improvement. Strategy 1 has similar performance (when 10 links are added) because the random selected nodes probably have low degree because in BA power law graphs there are a high number of nodes with few connections. The performance of strategy 4 seems to be close to that of strategy 2 where links are added between the minimum degree node and a random one. There are a high quantity of nodes with low degree and few nodes with high degree. The probability that a random node possesses a low degree node is high.

Thus, adding to add a link between two nodes with minimum degree is approximately the same as adding a link between the minimum degree node and a random one. See Figure 14.

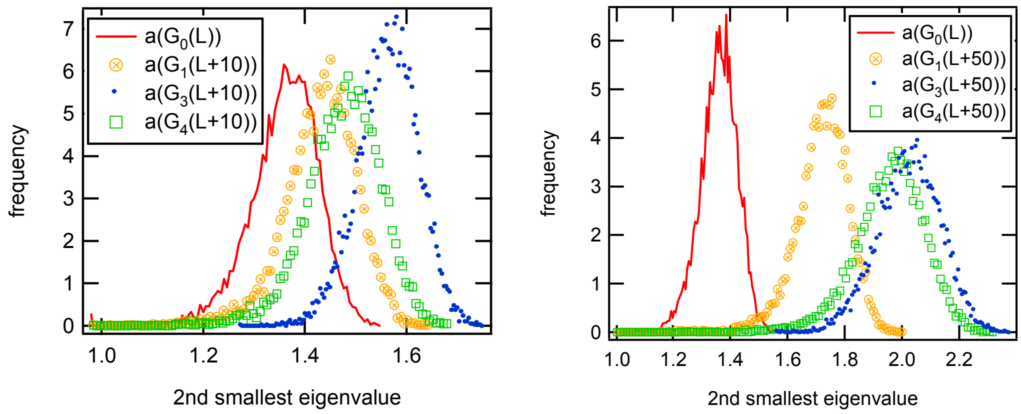


Figure 14. Algebraic connectivity in Barabasi-Albert model after adding 10 and 50 links with N=100 and m=3 using strategies 1,3 and 4.

5.1.3 Results of strategy 5

Strategy 5 involves adding a link between the maximum degree node and a certain number of nodes of the network. When 10 links are added to the network all of them are added between the maximum degree node and the rest of the nodes. However, when there is an addition of 50 links, depending on the link density, it could be impossible to add all of them to the maximum degree node. For this reason, the maximum degree node will withstand as much links as he can and then the others will be added between the second maximum degree node of the network and a random one and so on.

As you can see in the following graphs (Figure 15) strategy 5 performs similar to strategy 2, It does not improve too much the algebraic connectivity until the number of added links is high. The results after the addition of 50 links seem to be similar to strategy 3 and it is better than strategy 2 and 4.

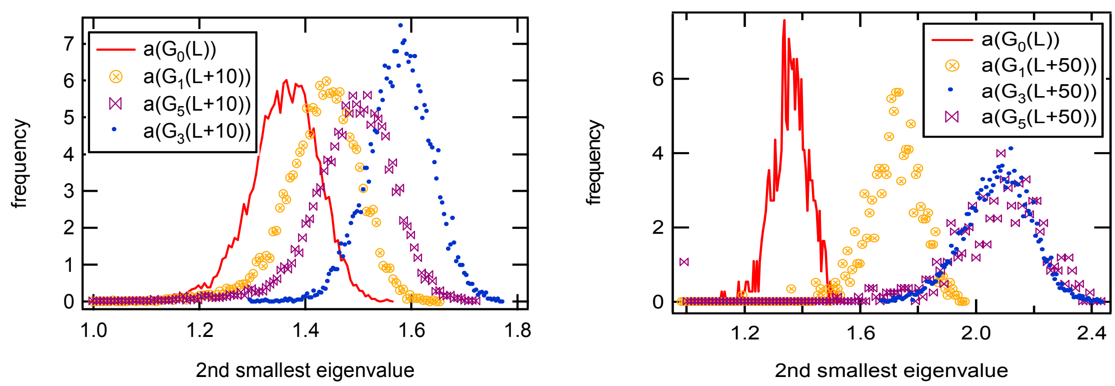


Figure 15. Algebraic connectivity in Barabasi-Albert model after adding 10 and 50 links with m=3 using strategies 1,3 and 5.

Finally, the best way to evaluate the overall performance of the strategies is making a comparison between all of them. Figure 16 shows the value of the algebraic connectivity after adding 10 and 50 links with all five initial strategies.

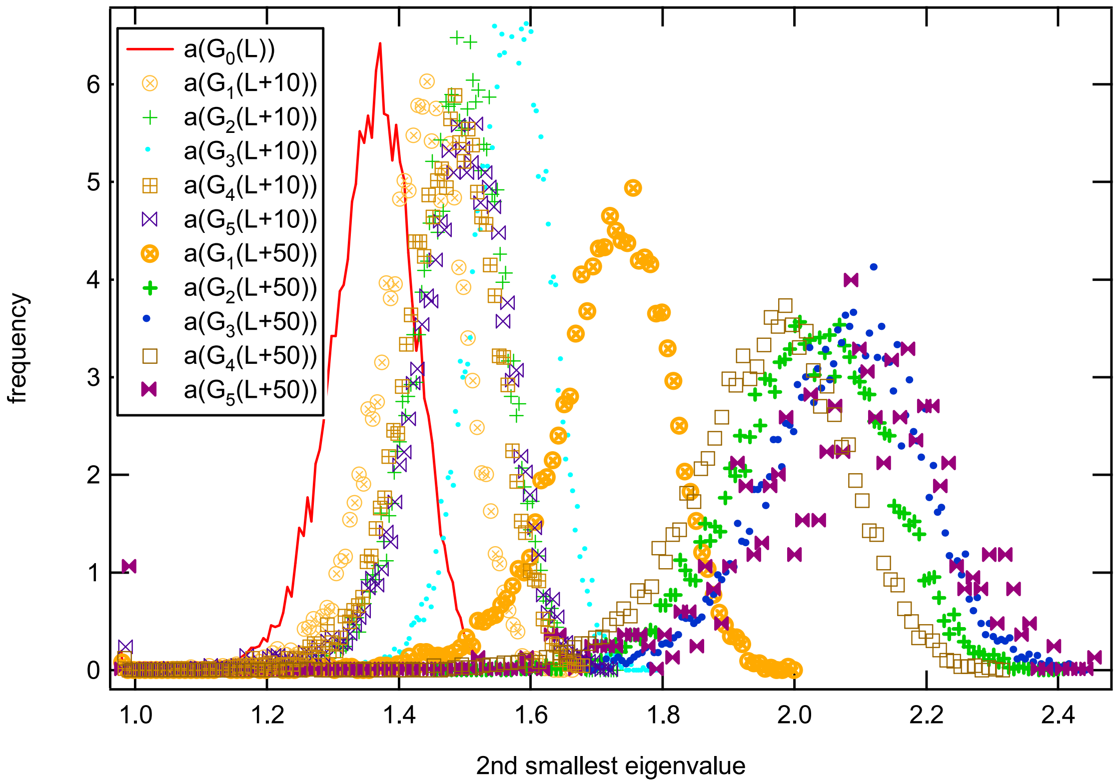


Figure 16. Five strategies comparison after adding 10 and 50 links. In a network with N=100 and m equal to 3. (Barabasi-Albert)

It is important to mention that the previous and next statements have to be corroborated with average values. This is due to the fact that although the histogram of a variable contains more information than the average of a random variable, some waves of the histograms are very close to each other and it is difficult to notice the concrete value by only having a look to them. From Figure 16 we can extract that as we have said before the *a(G)* increases proportionally as the number of links increases. All the strategies improve a(G) but the best ones in terms of increasing a(G) are strategy 3, 2 and 5.

We are going to check the exact difference between strategies obtaining the average algebraic connectivity and the average increase of algebraic connectivity with reference to the original network. Then we will be able to observe for example how after the addition of 10 links the difference between strategy 4 and 5 and between strategy 2 and 3 is noticeable whereas when we add 50 links the value of the algebraic connectivity is close among them.

5.1.4 Results of strategies 1, 2 and 3 with lower m

Furthermore, it is interesting to evaluate the performance of the strategies while changing the value of average degree *m* of the BA model. A lower *m* means that there is less links between the nodes so the network is tree-like. Concretely, we choose *m* =1.

The performance is similar to the previous simulation where m is equal to three. Strategy 3 performs better than the other two when the addition of links is 10. But as the number of links increases the difference between strategy 2 and 3 is smaller than the case of m = 3 (See Figure 17)

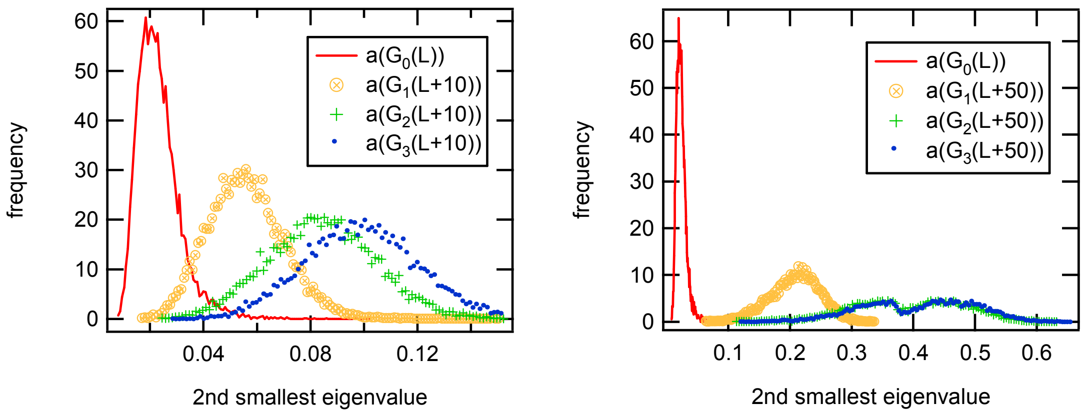


Figure 17.Algebraic connectivity in Barabasi-Albert model after adding 10 and 50 links with m=1 using strategies 1,2 and 3.

5.1.5 Average values

From the histograms we can extract that the algebraic connectivity depends on the number of added links as well as on the implemented link addition strategy.
Moreover, adding few links following the criteria of the maximum eigenvector distance implies a better performance in the network than adding a high amount of links by other strategies. For example, adding twenty links with strategy 3 or adding 50 with strategy 1, the algebraic connectivity is increased similarly. We will evaluate the performance of each strategy by the average increase of the algebraic connectivity after the link additions. Furthermore, we are also going to get the difference between the values after the link additions with reference to the original network.

We are going to investigate the average improvement in the algebraic connectivity with reference to the original network. The process to get the value is detailed below. First of all, we have to upload the data that we have previously used to print the waves and to create a new wave that is the difference between the two that we want to compare. A sequence of 100 binds is created. After that we assign to this wave the difference between the addition of 10 links using strategy 3 and no link addition in the same network (original network). Then, a new wave is created. This new wave is the needed one to print the previous result as an histogram. Only the average value of the increase is shown due to the high quantity of iterations and the difficulty of distinguishing between them.

It is important to mention that all the average values are computed based on the 10^4 iterations simulations.

The complete list of results can be consulted in Appendix B.

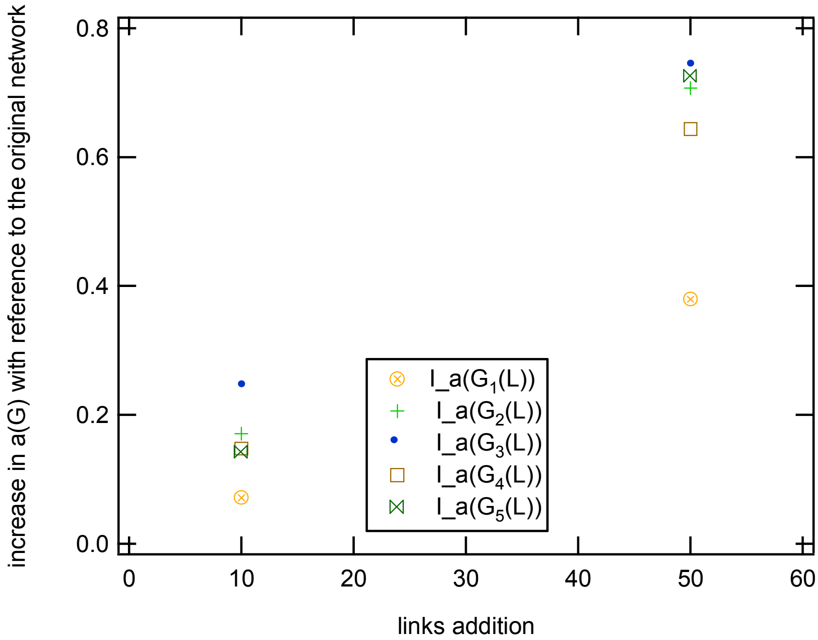


Figure 18. Improvements in a(G) after adding 10 and 50 links with all the strategies and with reference to the original network.

As shown in Figure 18 the increase of the algebraic connectivity by each strategy is on average between 0 and 0.8.

Moreover, as the number of added links increase, the improvement increases and the performance of strategy 2 and 5 get closer to strategy 3.We can also notice that as the number of added links increases the difference between the performance of strategy 1 and 2 is higher.

Due to the fact that it is important to check the proper working of the strategies we have also calculated the average eigenvalue of each strategy with each link addition strategies. The values shown in previous graphs are really close between them for this reason it is difficult to set statements about the simulations at first sight.

To obtain this information from IGOR pro the procedure is the following one: Analysis → Wave Stats and after that choosing the wave that we want to analyse and the range of the results (in our case between 0 and 10000 iterations). Below there is a screenshot (Figure 19) where this process is shown. You can see the selection of the waves, the obtained results that appear in the procedure window called Untitled and the graphs that are being analyzed.

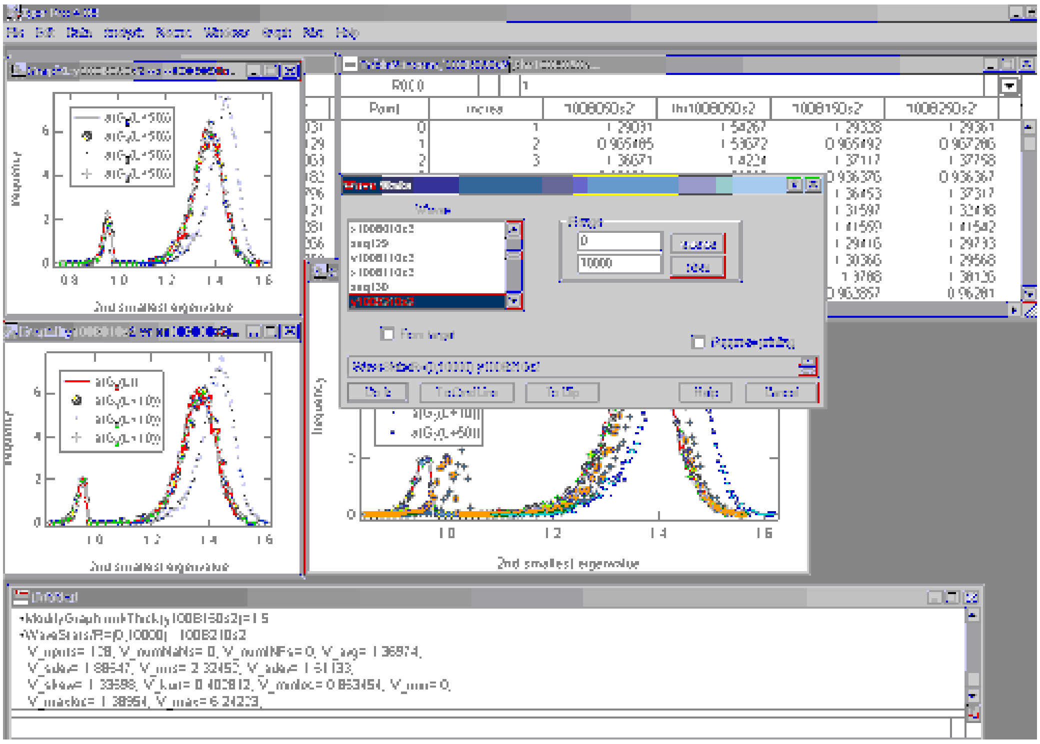


Figure 19. Acquisition of the waves' average value to corroborate the results.

As we have said before it is really useful to compare the average performance of the strategies when a different number of links are added. Figure 20 shows the average algebraic connectivity (second smallest eigenvalue) in relation to the number of link additions and the implemented strategy.

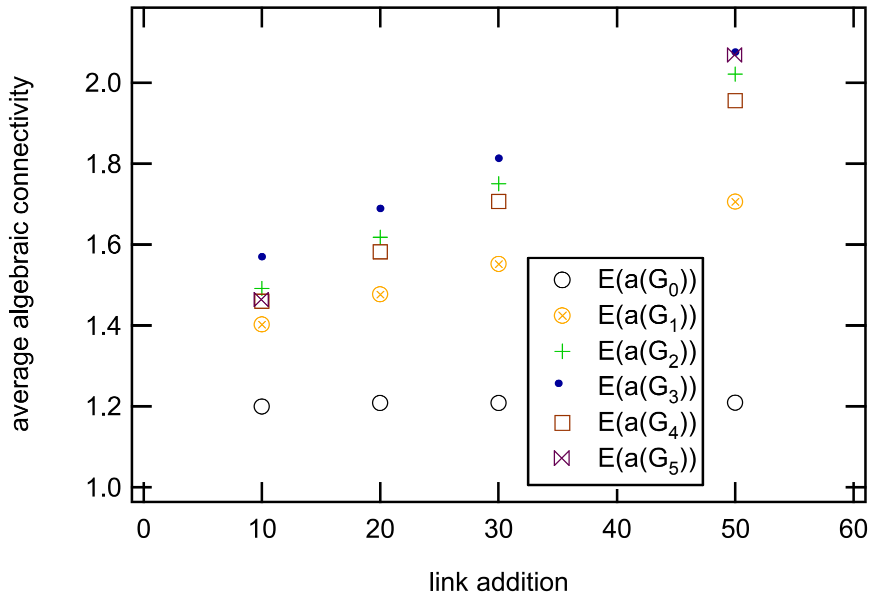


Figure 20. Average of the algebraic connectivity after the addition of links using different strategies (Barabasi-Albert model)

Figure 20 illustrates that *strategy 3 is the one that provides a higher a(G) followed by strategy 2.*

As we have noticed, when 50 links are added by strategy 5 or 3, the average value of *a(G)* is almost the same. Even strategy 2 is closer to strategy 3 as the number of added links increases.

Strategy 4 seemed to have behaviour almost equal to strategy 2 in the histogram but after calculating the average values we can observe how the performance is a little bit worse. On the other side, strategy 5 works well and strategy 1 is the least efficient way to increase the *a(G)* of the network. These results do not mean that strategy 1 is not advisable. For example, it has other advantages such as that it does not require any previous information about the nodes because they are chosen randomly.

Finally, as shown in Figure 18, the increase of algebraic connectivity by adding 50 links with e.g. strategy 3 is far less than 3 times the increase by adding 10 links. Thus, in order to achieve a higher increase in algebraic connectivity, a higher (than linear) cost or effort (in terms of the number of added links) is needed.

5.2 Erdös-Rényi Results

As in the previous section, we are going to calculate the algebraic connectivity using the six previously defined strategies.

5.2.1 Results of strategies 1, 2 and 3

First of all, we consider the Erdös-Rényi random graphs when the number of nodes of the network is N=100 and *p*= 0.4.

In Figure 21 we can observe that the performance of strategy 2 is really similar to that of strategy 3, which works better. Strategy 2 performs similarly to strategy 3 because we are connecting the nodes with less links of the network, so the selection is not random. It is important to remember that ER random graphs the distribution of links between nodes is homogeneous. Moreover, there are few nodes with few connections and the strategy selects one of them and with this situation the possibilities of having pendant nodes decrease.

Strategy 1 (random) does not generate almost any improvement because the links are added randomly. The algebraic connectivity improves because the number of links is higher than before but the enhancement is not noticeable.

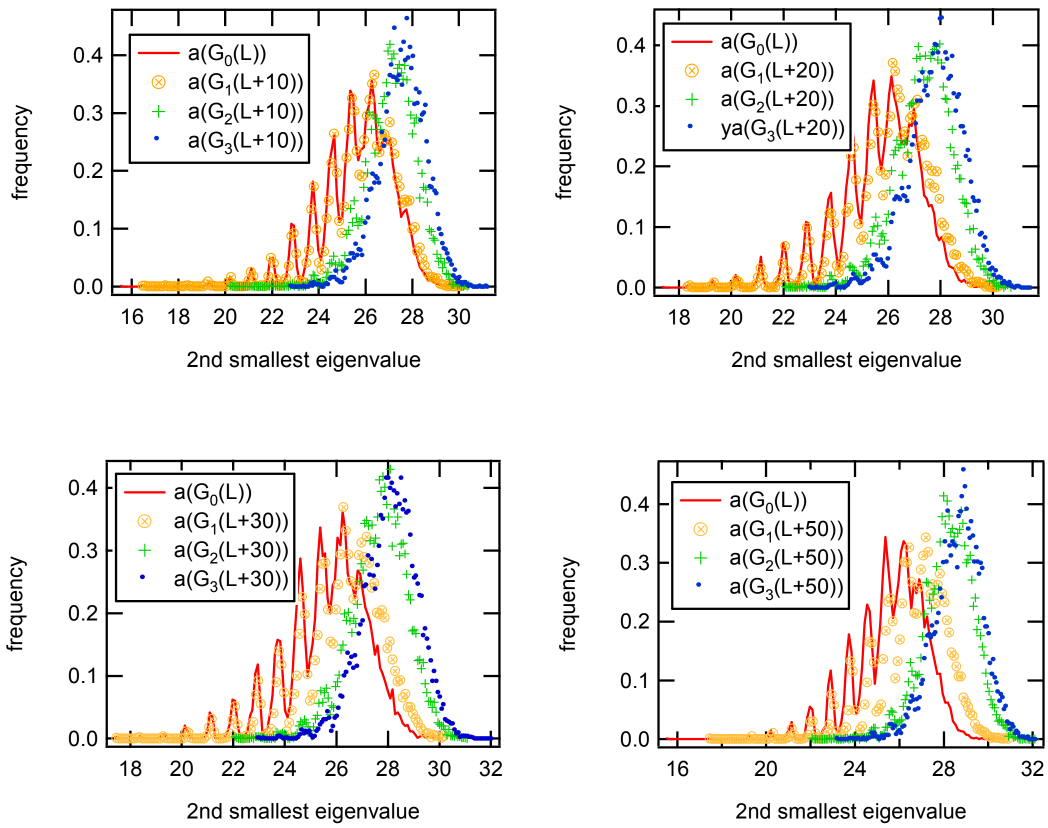


Figure 21. Algebraic connectivity in Erdös-Rényi model after adding 10, 20, 30 and 50 links with N=100 and p=0.6 using strategies 1,2 and 3.

5.2.2 Results of strategy 4

Now we are going to carry out the same simulations evaluating strategy 4. As it is noticeable in Figure 22, strategy 4 provides few improvements, the algebraic connectivity is almost the same as without applying any method or using strategy 1, only method 3 provides improvement in the algebraic connectivity. Because when we add a link between two nodes with the minimum degree we are not improving algebraic connectivity so much because they continue having low connections with the rest of the network.

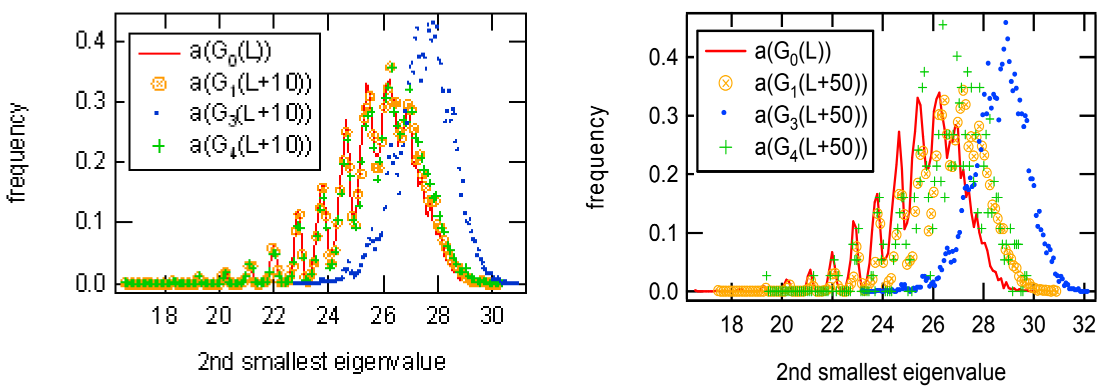


Figure 22. Algebraic connectivity in Erdös-Rényi model after adding 20 and 40 links with N=100 and p=0.6 using strategies 1,3 and 4.

5.2.3 Results of strategy 5

In our way to find a method that increase the algebraic connectivity, strategy 5 has been created. It lies in adding a link between the maximum degree node and another random node. As we have previously mentioned in Section 5.2.1.When 10 links are added to the network all of them are added between the maximum degree node and the rest of the randomly selected nodes. However, when there is an addition of 50 links it is impossible to add all of them starting from the same maximum degree node.

The unavailability of adding more links to a node is due to the selected *p* of the network, if link density *p* is lower there would be few links between nodes and probably it would be possible the addition of links only in one node. For this reason, the maximum degree node will withstand as much links as he can and then the others will be added between the second maximum degree node of the network and a random one and so on.

After 50 links being added the performance of strategy 5 is rather better than strategy 1. When 50 links are added the enhancement of *a(G)* is higher than with 10 links so as the number of links increases *a(G)* increases as well. This strategy works better than strategy 1 but worse than strategy 2 and 3. Then, it is more efficient, in terms of *a(G)* increase, to add a link between the minimum degree node and a random one than to add a link between the maximum degree node and a random one..

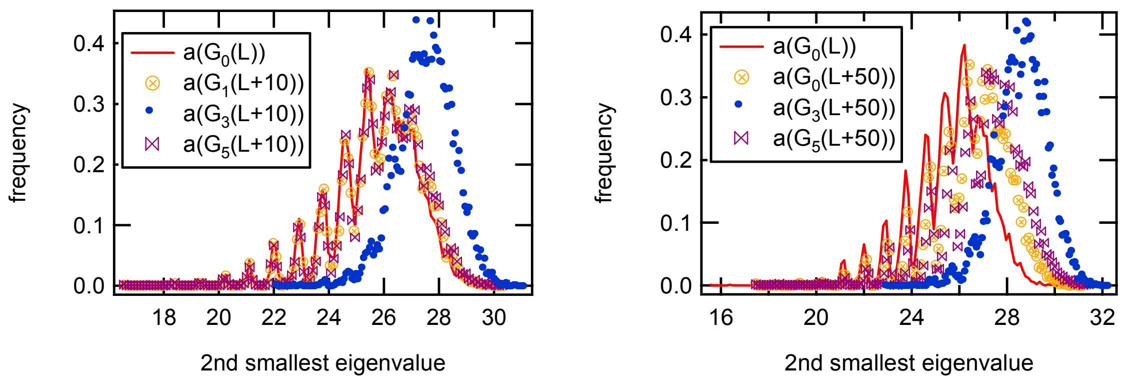


Figure 23. Algebraic connectivity in Erdös-Rényi model after adding 10 and 50 links with p equal to 0.6 using strategies 1,3 and 5.

Finally, as we have done in the previous model, the best way to evaluate the overall performance of the strategies is making a comparison between all of them. Figure 18 presents the histograms of the algebraic connectivity after adding 10 and 50 links with all five strategies. We can observe, the increase in the *a(G)* as the number of added links increases. Other information about the results has been expounded in previous graphs.

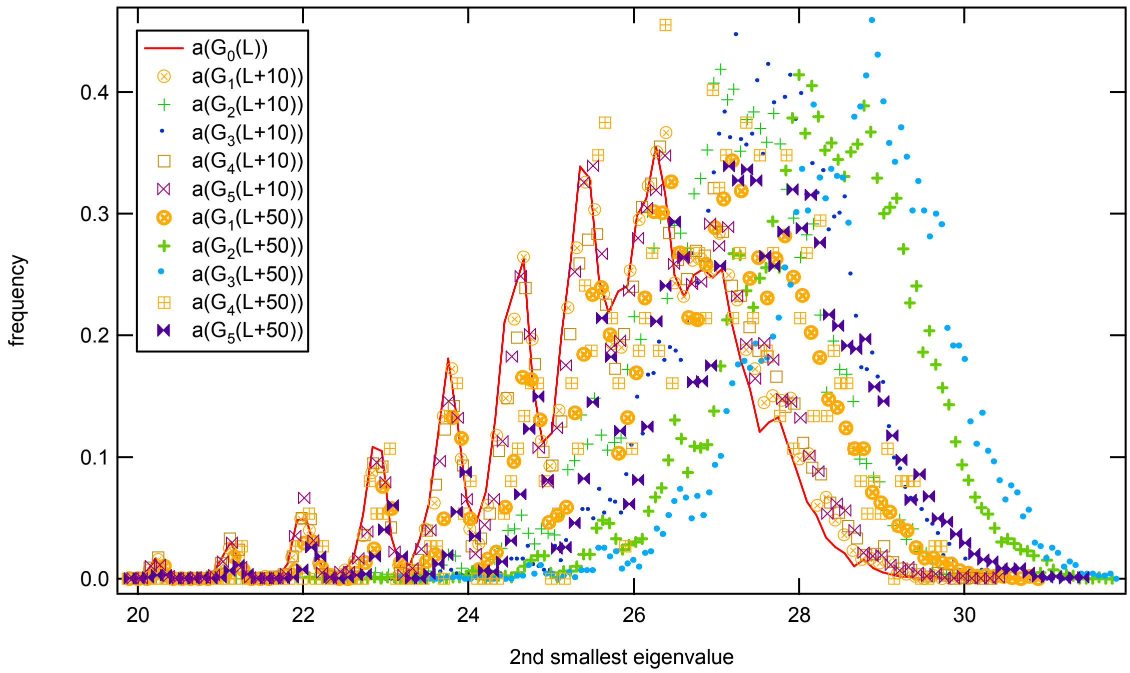
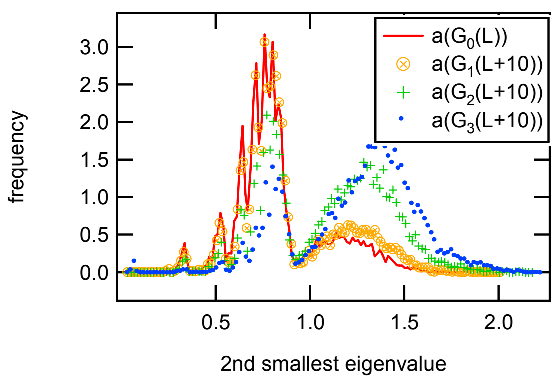


Figure 24. Comparison of the algebraic connectivity in Erdös-Rényi model with p equal to 0.6 and between the addition of 10 and 50 links using all the strategies.

5.2.4 Results of strategies 1, 2 and 3 with lower p

After doing these simulations, the next step is to set up a lower probability of link connectivity (*p*) in order to examine the influence of link density on the performance of our strategies. We set *p* equal to 0.06. In this case, the probability that a graph is disconnected is higher than that in previous simulations where p=0.4.

With this lower probability the results (Figure 25) shows that strategy 3 still work better than strategy 2. The effect of strategy 1 is noticeable when 50 links are added.



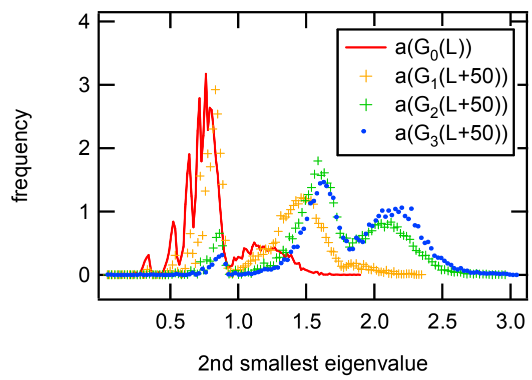


Figure 25. Algebraic connectivity in Erdös-Rényi model after adding 10 and 50 links with p equal to 0.06 using strategies 1,2 and 3.

5.2.5 Average values

We can extract from the graphical results that strategies 1 and 4 are not good in order to add links to the networks because the improvement in the *a(G)* is really low. Strategy 5 performs well when the number of added links is high.

As we have done in Barabasi- Albert results, the average second smallest eigenvalue has been calculated to provide a straightforward comparison among our strategies.

The procedures to get the increase in the algebraic connectivity with respect to the original network and the average value are the same as in Barabasi Albert model.

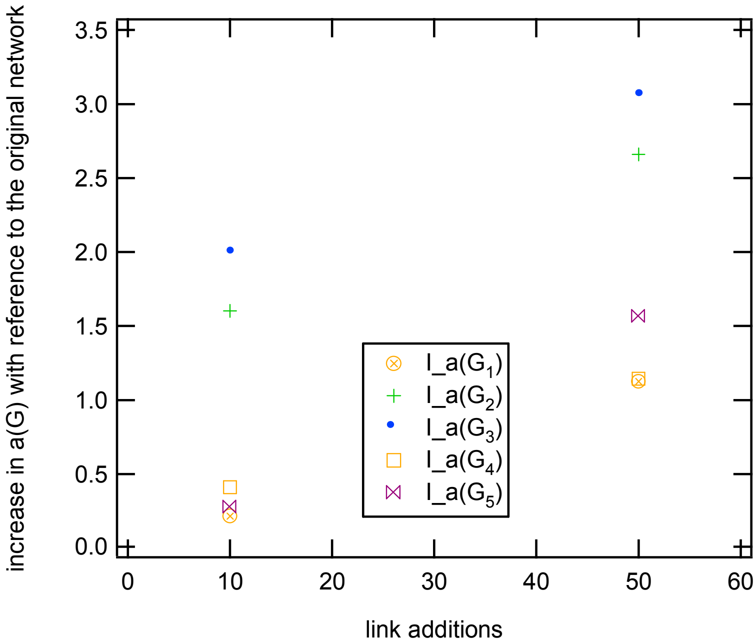


Figure 26. Improvement in a(G) after adding 10 and 50 links with all the strategies and with reference to the original network.

Next, the graph with the average algebraic connectivity is shown (Figure 27).

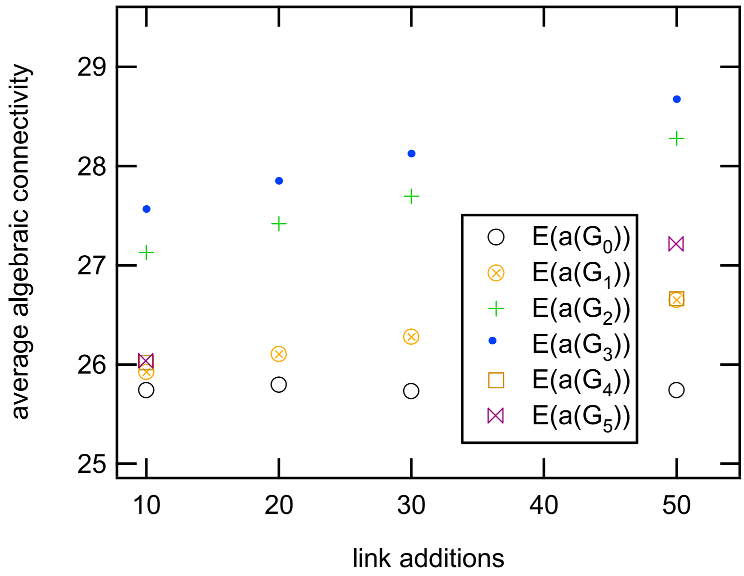


Figure 27. Average of the algebraic connectivity after the addition of links using different strategies (Erdös-Rényi model)

We can corroborate that as the number of links increases the average algebraic connectivity increases as well. Strategy 5 performs better than strategy 1 and 4 especially as the number of links increases, because we are always choosing the maximum degree node to start the link addition. Strategy 4 has almost the same performance as strategy 1. It only performs slightly better than strategy 1 when 10 links are added. However, when the number of links increases it tends to perform more or less equal than strategy 1. Finally, as in BA model the method that works better is number 3 followed by 2 and the one that performs worst is strategy 1.

5.3 Comparison of results between models

It is also interesting to compare the results between models. We can make a comparison between the results of Barabasi Albert where m=3 with the Erdös-Rényi where p= 0,06 since they possess the same link density: First equation guarantees the connectivity of the graph.

$$P_{Erdos-Renyi} = \alpha \frac{\log N}{N} = 3\frac{\log 100}{100} = 0.06$$
$$P_{Barabasi-Albert} = \frac{2m}{N-1} = 0.06 \rightarrow m = 3 \qquad (5.1)$$

Figure 28 and 29 shows how in the same conditions Barabasi works better because the value of the algebraic connectivity is higher than in Erdös-Rényi one. Only strategies 1, 2 and 3 are taken into account for the comparison but their performance has been previously evaluated. Strategies 4 and 5 were only simulated with a higher probability and adding a certain number of links and for this reason they are not taken into account in this comparison

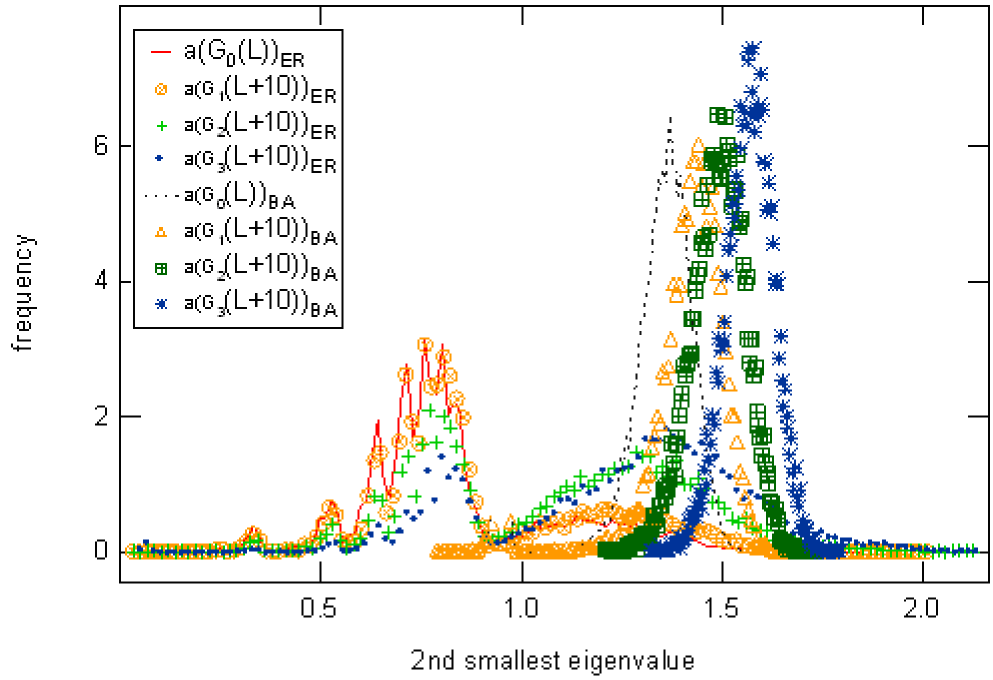


Figure 28. Comparison between the resuls of Barabasi-Albert and Erdös-Rényi models, after the addition of 10 links with strategies 1, 2 and 3

The performance evaluation of the different strategies has been done in previous sub-sections so here only the difference between models is considered.

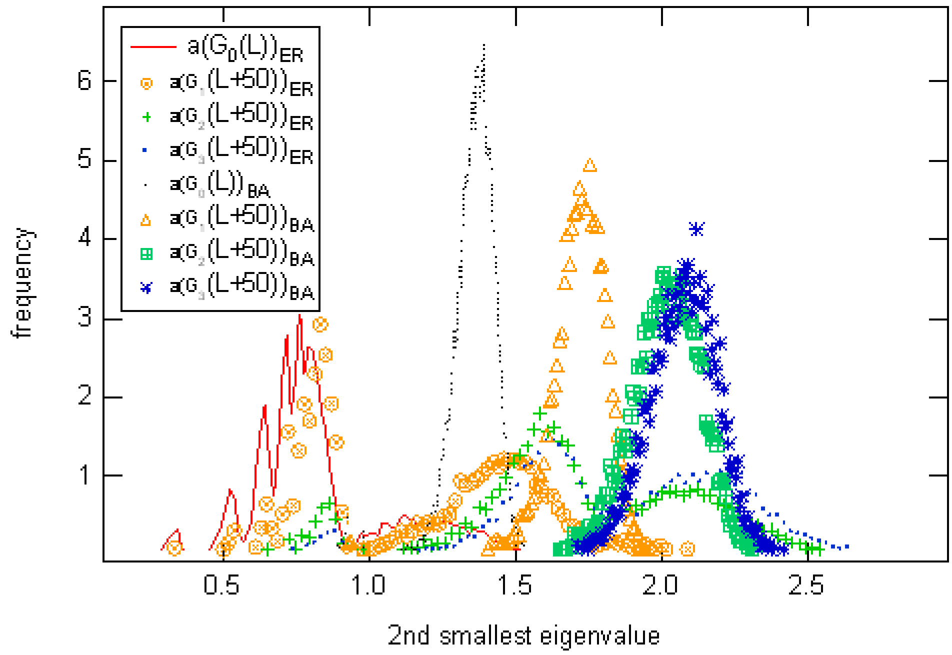


Figure 29. Comparison between the resuls of Barabasi-Albert and Erdös-Rényi models, after the addition of 50 links with strategies 1, 2 and 3

Finally, there is a comparison with the average values (Figure 30).

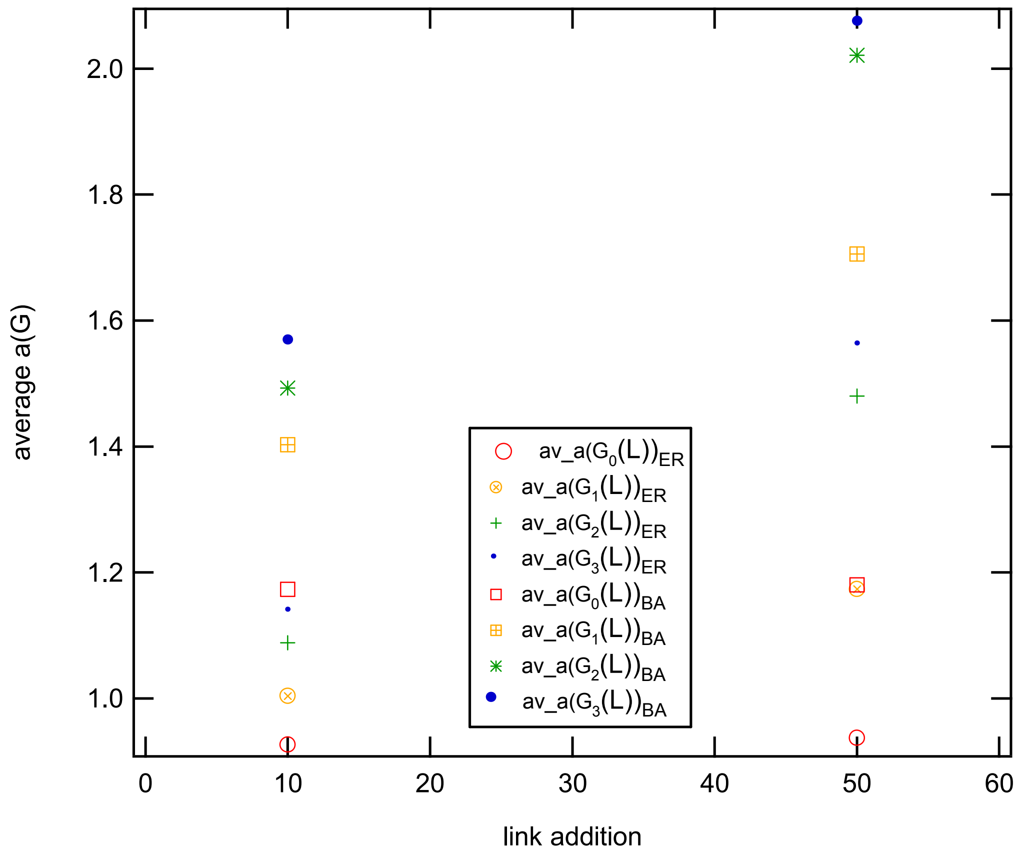


Figure 30. Average comparison of the algebraic connectivity after the addition of links using different strategies.

From the previous graph we can corroborate that the algebraic connectivity of the Barabasi-Albert model is generally higher than that of the Erdös-Rényi random graphs. Moreover, in ER model, the values of the algebraic connectivity using different strategies are closer between them. It is also important to see that after 10 added links being added with strategy 3, the performance of the ER model is still worse than the one of BA model without adding links (original network). On the other hand, adding 50 links the results change a little because strategy 3 and 2 using ER model have higher *a(G)* on average than strategy 1 and no strategy using or taking into account Barabasi-Albert model. As the number of added links increases, BA continues having a higher *a(G)* but ER is not as bad as with a low number of added links.

We can conclude saying that with the same number of nodes and links, the BA model has a higher algebraic connectivity than the ER model. However, after adding a small number of links properly to the ER model, the ER may outperform the original BA network model.

5.4 Link addition one by one

After doing all the simulations where the links are added all together and after that the *a(G)* calculation is computed, next step is to evaluate what happens when the links are added to the network one by one and the calculation is done after each link addition. We would like to see which the difference between the value of the algebraic connectivity after the addition of 10 and 50 links using both methods is.

This new method or strategy for link addition is going to be mentioned as strategy 6. It consists of adding a link between the nodes with the maximum algebraic distance. Strategy 3 added a set of links between pairs of nodes with the maximum algebraic distance and when the addition had finished the algebraic connectivity was calculated. On the other side, strategy 6 add links one by one, selects the pair of nodes with the maximum algebraic distance, add a link between them and then algebraic connectivity is calculated. This step is repeated till we reach the maximum number of links that we wanted to add.

Other strategies have the same performance adding the links all together or adding them one by one. The average algebraic connectivity is the same so that the performance is the same.

5.4.1 Barabasi-Albert

After adding 10 and 50 links to the network one, the histogram of the algebraic connectivity is shown in Figure 31.

The improvement in algebraic connectivity is lower when the links are added one by one than when they are attached once all together. Moreover, we can notice an enhancement in the algebraic connectivity between the addition of 50 and 10 links when the links are added once all of them. On the other hand, when the links are added one by one an improvement in *a(G)* as the number of links increases is not evident. We have evaluated average values to check this statement.

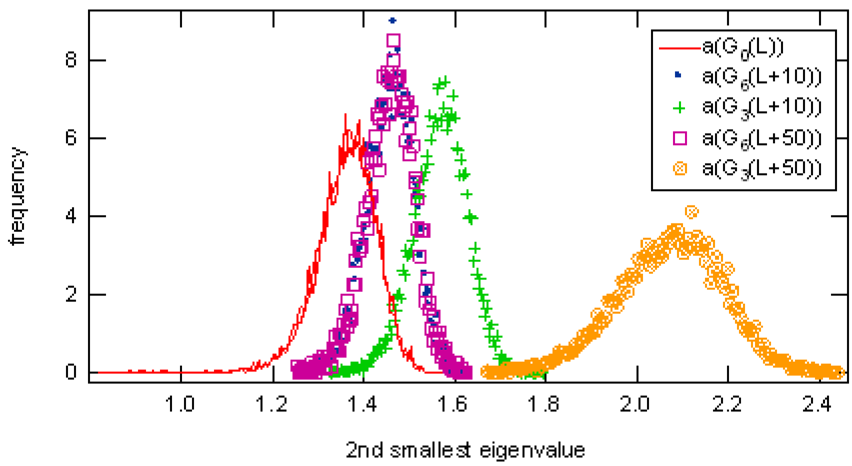


Figure 31. Algebraic connectivity in BA model after the addition of 10 and 50 links using strategies 3 and 6.

We have also compared the result of adding 10 links one by one with the addition of the same number of links but using strategies 1 and 2 and adding all of them together (or one by one because as we have previously mentioned the performance is the same). It is observable that the performance of strategy 6 is slightly better than random strategy and worse than strategy 2 (See Figure 32).

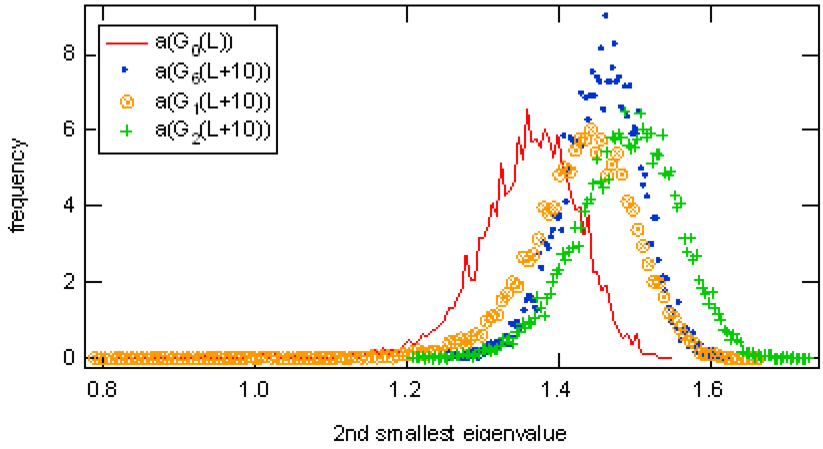


Figure 32. Algebraic connectivity in BA model after the addition of 10 links using strategies 1, 2 and 6. N=100 and m=3.

Average values are also considered to check previous statements. The average values have been calculated following the same procedure as before. The results are shown in Table 1.

Table 1. Average algebraic connectivity using different strategies (BA model)

Links \ Strategy	0	3	6
10	1,3293	1,56955	1,45727
50	1,3293	2,07355	1,45886

The average *a(G)* is higher with strategy 3 than 6. Therefore, it is better to add the links together and then calculate the *a(G)* .Also, the average *a(G)* after adding 50 links is higher than after the addition of 10 links. The improvement between both link additions is minimum so it is important to check the intermediate values to ensure that this difference is not due to numerical error (we can guarantee only two digit accuracy in our simulation result). It is verified that algebraic connectivity never decreases when more links are added to the network because all the intermediate average values from 0 to 50 links are considered.

On the other hand, Table 2 shows the increase in *a(G)* after link addition. The increase in algebraic connectivity is really low if we compare it with the one obtained with strategy 3.

Table 2. Average increase in the algebraic connectivity using different strategies (BA model)

Reference	Increase in algebraic connectivity
$a(G_6(L+10)) – a(G_0(L))$	0,137829
$a(G_6(L+50)) – a(G_0(L))$	0,139468
$a(G_6(L+50)) – a(G_6(L+10))$	0,00163889
$a(G_3(L+50)) - a(G_6(L+50))$	0,61469
$a(G_3(L+10)) - a(G_6(L+10))$	0,107902
$a(G_3(L+10)) - a(G_0(L))$	0,2479
$a(G_3(L+50)) - a(G_0(L))$	0,7455

5.4.2 Erdös-Rényi

In this subsection we investigate the adding of links one by one to the network using ER model.

In Figure 33 we can see how the performance of strategy 6 after the addition of 10 links is almost the same as that after adding 50 links. As in BA model the intermediate values of algebraic connectivity have been checked to ensure the correctness of the results. Using strategy 3, so that, adding the links all together the performance after adding 10 links is similar to one by one link addition However, when 50 links are added with strategy 3 the enhancement is higher than adding them with strategy 6. Nevertheless, using all the strategies there is an increase in *a(G)* over the original network.

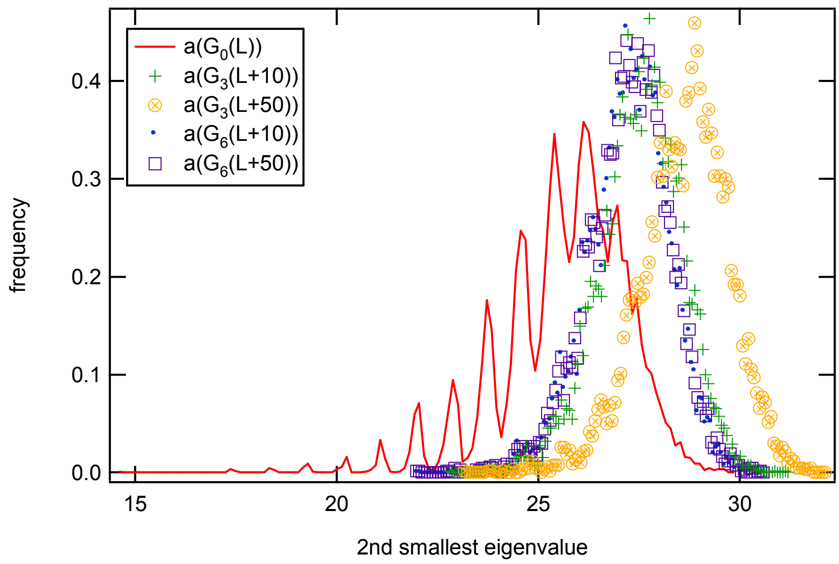


Figure 33. Algebraic connectivity in ER model after the addition of 10 and 50 links using strategies 3 and 6. (N=100 and p=0,04)

Comparing the performance with previously implemented strategies (1 and 2) we can see how the algebraic connectivity using strategy 6 is more or less the same as the one provided by strategy 2.

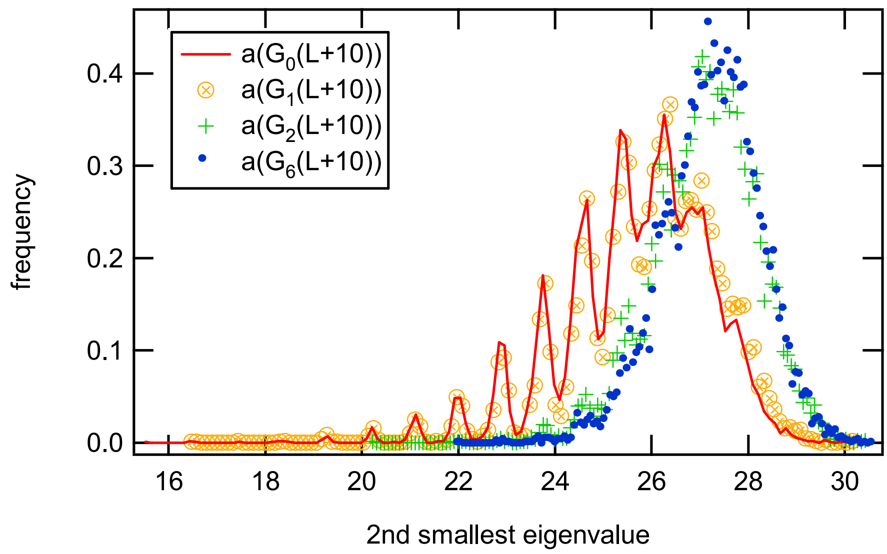


Figure 34. Algebraic connectivity in ER model after the addition of 10 and 50 links using strategies 1, 2 and 6.

As in previous model the average values have been calculated and the result is shown in Table 3. We can see how *a(G)* always increase with reference to the original network whatever the implemented strategy is. It is corroborated how the increase in the algebraic connectivity is higher using strategy 3 than using 6. And it is also noticeable how in strategy 6 after the addition of 10 or 50 links the average of *a(G)* is almost the same.

Table 3. Average algebraic connectivity using different strategies (ER model)

Links \ Strategy	0	3	6
10	25,7353	27,5698	27,2627
50	25,7353	28,6765	27,2929

Table 4. Average increase in the algebraic connectivity using different strategies (ER model)

Reference	Increase in algebraic connectivity
$a(G_6(L+10)) - a(G_0(L))$	1,78338
$a(G_6(L+50)) - a(G_0(L))$	1,8142
$a(G_6(L+50)) - a(G_6(L+10))$	0,0308148
$a(G_3(L+50)) - a(G_6(L+50))$	1,59022
$a(G_3(L+10)) - a(G_6(L+10))$	0,494163
$a(G_3(L+50)) - a(G_0(L))$	3,07735
$a(G_3(L+10)) - a(G_0(L))$	2,01329

We can see how there is more improvement using strategy 3 than 6 with the same number of links so it is better to add all the links all together. As the number of added links increases the difference between strategies is higher.

Moreover, we have checked how after the link addition of 10 and 50 links the enhancement in the original *a(G)* is noticeable. More added links improves the *a(G)* but as in previous model the increase here is really low.

To sum up, we can see how strategy 6 does not improve any more after adding more links. It is definitely the worst because the number of links that you add does not matter, after the addition of 10 links the performance is almost the same.

5.5 Other simulations

It would be interesting to evaluate which is the performance of link addition between the maximum and minimum degree nodes. We will try to provide information to know if this type of link addition performs better than the others.

I do not define this type of link addition as detailed as the other strategies because the objective is to briefly compare this type of adhesion with the others not to make an extensive research about its performance.

The addition consists of choosing the maximum degree node and the minimum degree node and adds a link between them. Next time, maximum degree node will continue being the same but minimum degree node can be another one. If there is a link between them we will change the maximum degree node for a random one.

This link addition is a mixture between strategy 5, where the maximum degree node is selected and strategy 2 where the addition between the minimum degree nodes and random ones is done. We will represent it as strategy 7 to distinguish it from other strategies.

As you can see in Figure 35, in ER model the performance is better than strategy 1 but worse than strategy 3. The performance is worse, on average, than strategy 2 where the addition is always between the minimum degree node and a random one (Table 5).

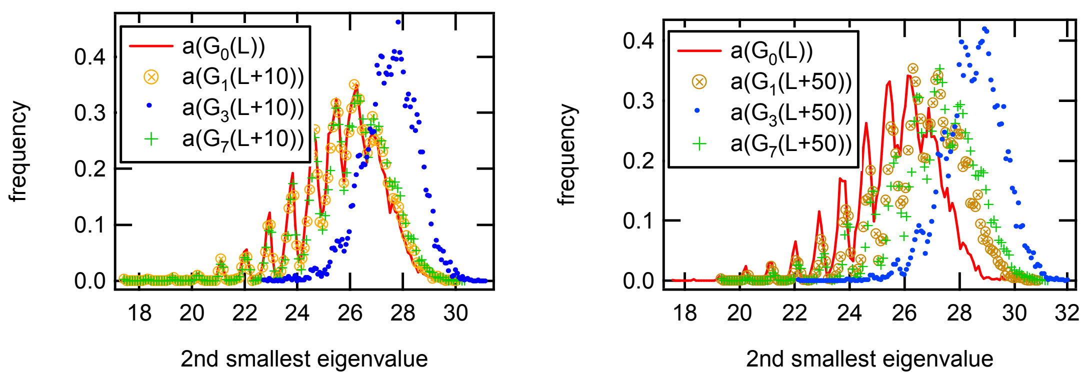


Figure 35. Evaluation of algebraic connectivity between maximum and minimum degree nodes (ER model N=100 and p=0.4)

Following tables (Table 5 and 6) show the average algebraic connectivity comparison. Strategy 7 performs almost the same as strategy 5 and also similar to strategy 4 when 10 links are added.

Table 5. Other simulations: Average algebraic connectivity using different strategies (ER model)

Links \ Strategy	0	1	2	3	4	5	7
10	25,7435	25,9261	27.1293	27.5698	26.0206	26.0317	26.0369
50	25,7438	26.6516	28.2791	28.6765	26.6624	27.2096	27.0969

The results of BA model (Figure 36) show how the performance of strategy 7 is better than strategy 1 and worse than strategy 3 when 10 links are added. On the other hand, when the number of added links increases the difference between strategy 7 and 3 is lower. The performance of strategy 7 in BA model is almost the same as strategy 2 with the same model as we can check on the average values that are shown in Table 6. As happened in ER, strategy 7 performs almost like strategy 5.

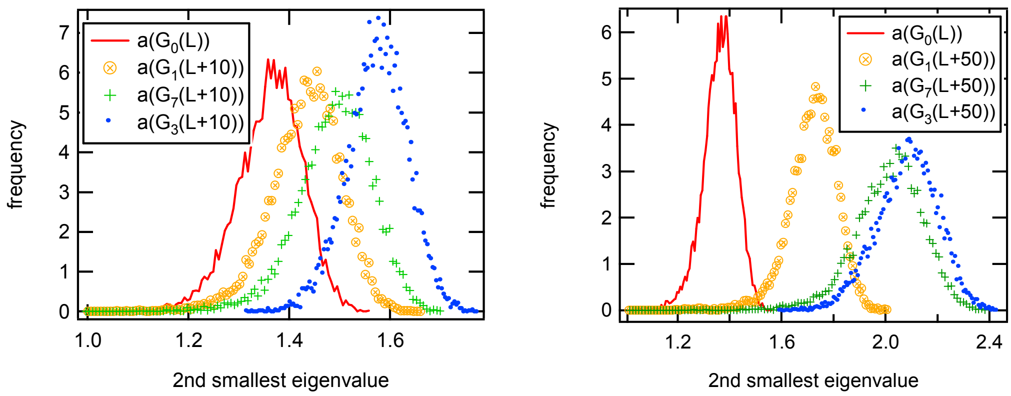


Figure 36. Evaluation of algebraic connectivity between maximum and minimum degree nodes (BA model N=100 and m=3)

Table 6. Average algebraic connectivity using different strategies (BA model)

Links \ Strategy	0	1	2	3	4	5	7
10	1.17299	1.22618	1.46942	1.55796	1.2549	1.46334	1.46201
50	1.2088	1.44635	2.02123	2.07605	1.91561	2.06719	2.0067

We can conclude saying that it is not necessary to implement this strategy and carry out all the simulations because it has a similar performance as strategy 2 in BA model and similar to strategy 5 in both models. It does not provide any new useful information.

CHAPTER 6. CONCLUSIONS

In this chapter we are going to show the main conclusions obtained after the development of the project.

After doing all the simulations and obtaining all the data now we will try to summarize the obtained conclusions. Next statements have been enunciated before in previous subsections. We have obtained the histograms with the complete results and after that we have done the calculation of the average values.

About BA we can conclude that:

- Algebraic connectivity increases as the number of links increases
- Strategy 1 is the one that performs worse but it does not require local information.
- Strategy 3 is the one that performs the best.
- Strategy 2, 4 and 5 performs similarly when 10 links are added.
- Strategy 2, 5 and 3 has a similar improvement in *a(G)* when 50 links are added.

Referring to ER we can sum up telling that:

- Strategy 3 is the one that performs the best. It is followed by strategy 2.
- Strategy 1 does not remarkably improve the algebraic connectivity of the original network until the number of added links is high.
- Strategy 4 provides little improvement especially when 50 links are added, because we are adding links between nodes with similar degree so the enhancement is similar to strategy 1.
- Strategy 5 works almost like strategy 4 when 10 links are added and better than 4 when the added links are 50.

About the comparison between ER and BA model, we can state that:

- Algebraic connectivity in BA model is higher than in ER model, even taking into account the original network algebraic connectivity before link additions.
- In ER, the values of algebraic connectivity using different strategies are closer between them than in BA model.

Concerning the difference between the addition of links one by one (strategy 6) or all together (strategy 3), we can notice how after adding the links once the increase in the algebraic connectivity is higher than using one by one addition.
Another drawback of the strategy 6 is that there is almost no difference between the addition of 50 and 10 links because the improvement with reference to the original network is similar in both cases. For this reason, the number of links that you add does not matter, after the addition of 10 links the performance is almost the same. In conclusion, strategy 6 provides more or less the same increase in *a(G)* as strategy 2. But it is the worst one because even increasing

the number of links of the number the algebraic connectivity does not increase anymore.

Finally, we can conclude with strategy 7 saying that after adding a link between a high and a low degree node the algebraic connectivity increases but it performs almost like strategy 5 (in ER and BA model). With BA model the results are also similar to the ones of strategy 2.

CHAPTER 7. FUTURE RESEARCH

Although an extensive work has been done, there are points which are not considered because of the lack of time. This means that, the project can be extended.

The aim of this chapter is to detail all the aspects that can be taken into account in the future. With all this information another researcher or institutions who are interested in this project, can continue developing it.

Further work comprises studying the algebraic connectivity in other complex network models and adding links with other strategies. This would improve our global understanding of whether the algebraic connectivity is a good way to measure robustness. A new model that can be taken into account is a small world graph of Watts-Strogatz. This model describes the fact that, despite the large size of the underlying network topology, in most complex networks there is a relatively short path between any two nodes [16]. Concerning the design of new strategies, other features of the network can be taken into account, for example, betweenness (the number of paths in the network that pass through a node) or clustering (the degree to which nodes tend to cluster together).

Moreover, it would be important to investigate how the algebraic connectivity is affected by various structural changes of the network graphs. Changes such as removing nodes or rewiring them could be studied. On the other hand, the same strategies can be simulated while changing the size of the network or links density parameters to evaluate the performance.

CHAPTER 8. ACKNOWLEDGEMENTS

Firstly, I would like to thank Huijuan Wang for following the project step by step and day by day and helping me to achieve the goals. Also to Javier Martin for helping me with computing problems

Furthermore, I would like to thank my family and friends for supporting me at every moment.

CHAPTER 9. BIBLIOGRAPHY

[1] NEWMAN,M.E.J.:The structure and function of complex networks, Departament of Physics, University of Michigan USA.

[2]FIEDLER,M.: Algebraic connectivity of graphs, Czechoslovak Mathematical Journal 23, pp.298-305 (1973)

[3]GUO,J.: The algebraic connectivity of graphs under perturbation, China University of Petroleum.

[4] JAMAKOVIC,A.-VAN MIEGHEM.P.: On the robustness of complex networks by using the algebraic connectivity, networking 2008,LNCS 4982 pp 183-194, (2008).

[5] ERDÖS,P- RÉNYI,A.: On random graphs. Publicationes Mathematicae 6, 290-297 (1959).

[6] Y.FAN, On spectral integral variations of graphs, Linear and Multilinear Algebra 50 (2002) 133-142

[7] S.BARIK, S.PATI, On algebraic connectivity and spectral integral variations of graphs, Linear Algebra Appl. 397 (2005) 209-222

[8] K.CH.DAS, The Laplacian spectrum of a graph, Computers & Mathematics with Applications, Volume 48, Issues 5-6 , September 2004, Pages 715-724

[9] JAMAKOVIC.A – UHLIG,S.: On the relationship between the algebraic connectivity and graph's robustness to node and link failures, Next Generation Infraestructure Program.

[10] BARABASI,A.-L.AND ALBERT,R.: Emergence of scaling in random networks, Science 286, pp.509-512 (1999)

[11] JUHAZ,F.: The asymptotic behavior of Fiedler's algebraic connectivity for random graphs, Discrete Mathematics 96, 59-63 (1991)

[12] WANG.H-VAN MIEGHEM.P.: Algebraic connectivity optimization via link addition, Bionetics'08, (2008)

[13] AOYAMA,D.: Maximum algebraic connectivity augmentation is NP-HARD, Operations Research Letters 36, pp 677-679 (2008)

[14] JAMAKOVIC,A-UHLIG,S.: Influence of the network structure on robustness, IEEE (2007).

[15] KANOVSKY,I.: Complex Networks Clustering and Links Correlation, Max Stern Academic College of Emek Yezreel,

[16] WATTS,D.J. AND STROGATZ,S.H.: Collective dynamics of small-world networks, Nature 393, pp.440-442 (1999)

IGOR Pro manual guide. WaveMetrics, Inc (version 4)

Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

APPENDIX

TITLE: Robustness optimization via link additions

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management (EPSC)

AUTHOR: Mari Carmen Sánchez Martínez

DIRECTOR: Piet Van Mieghem- Fernando Kuipers

SUPERVISOR: Huijuan Wang – Javier Martín

DATE: 13-07-2009

APPENDIX A.HISTOGRAM GENERATION

In order to create an histogram that shows the evolution of the algebraic connectivity in the network we need to compile the following code in the procedure window of IGOR pro.

```
#pragma rtGlobals=1		// Use modern global access method.
Function	Funchistogram(src,xdstname,dstname,numbins,seq)	//making histogram from source wave, histogram $dstname vs $xdstname
		Wave src,seq
		Variable numbins
		string dstname,xdstname// input is string

		Variable V_min,V_max
		make/N=(numbins) $xdstname,$dstname
		wavestats/Q src
		histogram/B={V_min,(V_max-V_min)/numbins,numbins} src,$dstname
		wave x=$xdstname
		wave y=$dstname
		y=y/V_npnts/((V_max-V_min)/numbins)
		x=V_min+(seq-1)*(V_max-V_min)/numbins
End
Function Funchistogramdiscrete(src,xdstname,dstname)	//making histogram from source wave, histogram $dstname vs $xdstname
		Wave src
		string dstname,xdstname// input is string

		Variable V_min,V_max,numbins
		wavestats/Q src
		numbins=V_max-V_min+1
		make/N=(numbins) $xdstname,$dstname
		histogram/B={V_min,1,numbins} src,$dstname
		wave x=$xdstname
		wave y=$dstname
		y=y/V_npnts
		x=p+V_min
End

Window Table12() : Table
	PauseUpdate; Silent 1		// building window...
	Edit/W=(5.25,41.75,577.5,432.5) seq106,seq107,seq108
EndMacro

Window Graph16() : Graph
	PauseUpdate; Silent 1		// building window...
	Display /W=(5.25,41.75,719.25,512.75) yseq501 vs xseq501
	AppendToGraph yseq502 vs xseq502
	AppendToGraph yseq503 vs xseq503
	AppendToGraph yseq504 vs xseq504
	AppendToGraph yseq505 vs xseq505
```

```
	AppendToGraph yseq506 vs xseq506
	AppendToGraph yseq507 vs xseq507
	AppendToGraph yseq508 vs xseq508
	AppendToGraph yseq509 vs xseq509
	AppendToGraph yseq510 vs xseq510
	ModifyGraph
mode(yseq502)=3,mode(yseq503)=3,mode(yseq504)=3,mode(yseq505)=3,mod
e(yseq506)=3
	ModifyGraph
mode(yseq507)=2,mode(yseq508)=3,mode(yseq509)=3,mode(yseq510)=3
	ModifyGraph
marker(yseq503)=5,marker(yseq504)=6,marker(yseq505)=4,marker(yseq506)=
28
	ModifyGraph
marker(yseq508)=11,marker(yseq509)=14,marker(yseq510)=8
	ModifyGraph lSize(yseq507)=2
	ModifyGraph
rgb(yseq502)=(0,52224,0),rgb(yseq503)=(65280,0,52224),rgb(yseq504)=(0,640
0,26112)
	ModifyGraph
rgb(yseq505)=(65280,43520,0),rgb(yseq506)=(0,0,39168),rgb(yseq507)=(6400,
0,13056)
	ModifyGraph rgb(yseq509)=(0,43520,65280)
	ModifyGraph tick=2
	ModifyGraph mirror=1
	Label left "frequency"
	Label bottom "improvement in the algebraic connectivity"
	SetAxis left 0,23.8836459119173
	SetAxis bottom -0.0143779160493056,1.02893090248108
	Legend/N=text0/J/A=MC/X=40.12/Y=28.75 "\\s(yseq501)
yseq501\r\\s(yseq502) yseq502\r\\s(yseq503) yseq503"
	AppendText "\\s(yseq504) yseq504\r\\s(yseq505) yseq505\r\\s(yseq506)
yseq506\r\\s(yseq507) yseq507\r\\s(yseq508) yseq508"
	AppendText "\\s(yseq509) yseq509\r\\s(yseq510) yseq510"
EndMacro
```

APPENDIX B. A(G) IMPROVEMENT (numerical results)

Difference between the a(G) after the addition of 10 and 50 links using the five defined strategies taking as a reference the original network. Here there are only the 100 first values but the average has been calculated the 10000 iterations.

Barabasi-Albert

10 links				
$a(G_1)$-$a(G_0)$	**$a(G_2)$-$a(G_0)$**	**$a(G_3)$-$a(G_0)$**	**$a(G_4)$-$a(G_0)$**	**$a(G_5)$-$a(G_0)$**
0.064789	0.102848	0.185038	0.129973	0.085672
0.088216	0.117764	0.163922	0.102929	0.18016
0.033388	0.090269	0.126153	0.009133	0.105836
0.053997	0.113898	0.138261	0.148045	0.189185
0.050108	0.089417	0.111395	0.012725	0.106892
0.115913	0.198162	0.200128	0.448529	0.124793
0.074331	0.121228	0.202458	0.076371	0.124574
0.063164	0.079821	0.260201	0.126422	0.092196
0.098739	0.123497	0.212244	0.080926	-0.068123
0.001594	0.605168	0.616102	0.577568	0.709238
0.046904	0.142851	0.296065	0.086183	0.133615
0.102369	0.184336	0.194775	0.059915	0.093158
0.028672	0.15418	0.228251	0.099472	0.12211
0.057627	0.071915	0.276476	0.177277	0.246788
0.02896	0.055737	0.194789	0.119894	0.183039
0.04594	0.056205	0.137789	0.045209	0.096999
0.032199	0.122466	0.132005	0.063998	0.082632
0.026671	0.099507	0.167878	0.130171	0.128445
0.048113	0.07886	0.12802	0.110563	0.07262
0.057073	0.122153	0.168065	0.269573	0.138597
0.063317	0.128677	0.134177	0.234273	0.223018
0.085946	0.110016	0.291224	0.083722	0.070949
0.066276	0.124978	0.158656	0.041214	0.142966
0.039972	0.098478	0.252013	0.038758	0.028219
0.145453	0.184305	0.299147	0.09023	0.046847
0.059035	0.214672	0.262793	0.193412	-0.315573
0.508144	0.563478	0.564989	0.374077	0.521826
0.052297	0.121038	0.380545	0.276607	0.120995
0.088505	0.13852	0.200983	0.119897	0.192369
0.078798	0.093054	0.179525	0.11361	0.022116
0.059261	0.107732	0.125538	0.045341	0.099549
0.031531	0.058278	0.256075	0.10286	0.185189
0.080312	0.136963	0.188776	0.037076	0.114383
0.015395	0.308167	0.316811	0.278432	0.240174
0.075637	0.117795	0.133316	0.073728	0.163659
0.05624	0.179461	0.181217	0.167052	0.038702

0.047494	0.091222	0.140768	0.137052	0.143009
0.07251	0.166372	0.207804	0.249403	0.109725
0.082933	0.149615	0.152605	0.069128	-0.120164
0.000781	0.599665	0.621666	0.578799	0.570167
0.062731	0.11484	0.226881	0.193079	0.086027
0.079084	0.093219	0.156064	0.104753	0.080463
0.1238	0.19276	0.234366	0.230017	0.208665
0.079912	0.15904	0.188008	0.112892	0.022505
0.043232	0.131599	0.185688	0.088649	0.089472
0.083479	0.095975	0.239281	0.129208	0.107752
0.023316	0.067105	0.213959	0.187124	0.186806
0.081686	0.094967	0.194881	0.065964	0.118243
0.047095	0.071032	0.267415	0.085094	0.152596
0.031862	0.04581	0.136559	0.146679	0.094809
0.028788	0.049424	0.140674	0.109454	0.251279
0.001389	0.571696	0.636548	0.474665	0.563813
0.056794	0.108008	0.171341	0.246801	0.360896
0.18009	0.225689	0.392258	0.245854	0.184996
0.029398	0.166587	0.179487	0.019673	0.07858
0.018958	0.152106	0.200556	0.123372	0.004767
0.494319	0.530488	0.541121	0.547724	0.600718
0.013525	0.05333	0.285456	0.077721	0.429591
0.080527	0.098906	0.196073	0.000573	0.199796
0.055998	0.119941	0.293125	0.131025	0.230455
0.046108	0.085098	0.108699	0.039885	0.176415
0.04151	0.061214	0.186572	0.037571	0.104179
0.07429	0.078545	0.146172	0.031532	0.064578
0.052983	0.069764	0.199467	0.216886	0.367513
0.089347	0.174831	0.229806	0.060574	0.051803
0.190486	0.225536	0.235076	0.169695	0.080697
0.126354	0.142765	0.153851	0.083784	0.079963
0.066264	0.17849	0.217102	0.10882	0.224781
0.086091	0.104671	0.112863	0.129639	0.126094
0.035783	0.048042	0.258647	0.062101	0.071063
0.107369	0.185565	0.202795	0.222387	0.252884
0.00066	0.467188	0.473948	0.492135	0.552817
0.054238	0.095144	0.222773	0.232713	0.009211
0.047891	0.067867	0.307817	0.129174	0.259446
0.100333	0.134814	0.22652	0.059029	0.296877
0.000734	0.509991	0.533094	0.475158	0.569504
0.057583	0.137681	0.170667	0.203306	0.152353
0.082521	0.099783	0.208902	0.105355	0.112884
0.028756	0.037892	0.20546	0.044999	0.270869
0.06249	0.11758	0.191251	0.186748	0.017297
0.042818	0.134516	0.151219	0.104681	0.067206
0.000969	0.384726	0.626175	0.616562	0.462511
0.003017	0.410236	0.455197	0.519189	0.349554
0.124231	0.185153	0.233401	0.153214	0.284458
0.088783	0.116839	0.189543	0.012824	0.116597
0.001141	0.569126	0.609684	0.531501	0.638812

0.178747	0.215767	0.246246	0.116389	0.184845
0.101136	0.144968	0.197544	0.016706	0.232051
0.071625	0.140673	0.371389	0.283282	0.136757
0.065916	0.109023	0.172169	0.012308	0.11949
0.10974	0.179899	0.195081	0.09829	0.042868
0.092536	0.18058	0.19445	0.071678	0.008378
0.037628	0.13487	0.165794	0.047616	0.018434
0.021179	0.243222	0.348168	0.160799	0.300199
0.001165	0.601293	0.632435	0.464899	0.544066
0.065697	0.126538	0.132244	0.217682	0.063435
0.051499	0.133272	0.21723	0.160121	0.070199
0.145216	0.193531	0.371165	0.082977	0.225224
0.095571	0.169788	0.258137	0.194907	0.063795
0.068982	0.095228	0.171195	0.14433	0.098486
				0.085623
Average				
0.0716397	0.17057	0.247983	0.147647	0.141598

50 links				
$a(G_1)$-$a(G_0)$	**$a(G_2)$-$a(G_0)$**	**$a(G_3)$-$a(G_0)$**	**$a(G_4)$-$a(G_0)$**	**$a(G_5)$-$a(G_0)$**
0.338253	0.706483	0.737972	0.667289	0.704136
0.006096	1.13963	1.16061	0.650502	1.25304
0.316941	0.690826	0.69955	0.722039	0.662007
0.44168	0.722682	0.731303	0.379677	0.52259
0.247295	0.7357	0.743347	0.715548	0.803963
0.544532	0.746758	0.753172	0.756802	0.446455
0.230825	0.73548	0.741881	0.729101	0.242111
0.36653	0.626358	0.659111	0.616636	0.745648
0.277587	0.512373	0.530608	0.733461	0.810181
0.417814	0.599342	0.617885	0.667595	0.753339
0.327554	0.888231	0.901538	0.772938	0.220478
0.259399	0.659134	0.788364	0.644592	0.880722
0.336966	0.74455	0.75102	0.534064	0.771292
0.386746	0.616118	0.625096	0.404852	0.599894
0.458021	0.681994	0.746847	0.596977	0.624359
0.353909	0.774282	0.776046	0.723676	0.952151
0.314255	0.514578	0.648428	0.699469	0.587054
0.337293	0.70639	0.717926	0.521345	0.722277
0.307712	0.658811	0.668762	0.386822	0.517897
0.676702	1.09627	1.10284	1.08556	1.03572
0.243406	0.578036	0.661308	0.686951	0.573058
0.419109	0.684301	0.689753	0.500066	0.834151
0.251473	0.436983	0.444578	0.716815	0.850767
0.302859	0.574223	0.590315	0.421311	0.528745
0.431022	0.586843	0.593892	0.409287	0.704894
0.455613	0.756582	0.875344	0.543962	0.363855
0.468934	0.727786	0.775247	0.504263	0.807223
0.305115	0.51153	0.521374	0.799269	0.827951
0.537247	0.805622	0.811074	0.662275	0.452036
0.285344	0.746946	0.785226	0.643015	0.838754

0.529196	0.641208	0.64238	0.575534	0.521578
0.438378	0.849221	0.85088	0.646995	1.16165
0.394004	0.736777	0.744938	0.533091	0.73176
0.283542	0.698676	0.717928	0.832595	0.827214
0.374429	0.753633	0.756584	0.640555	0.80645
0.381242	0.51026	0.51197	0.592501	0.261023
0.341636	0.513463	0.72985	0.770726	0.874758
0.222021	0.77123	0.774699	0.779044	0.54944
0.319831	0.740821	0.752542	0.575117	0.46103
0.578815	0.944952	1.00712	0.916654	1.00416
0.485162	0.919089	0.957947	0.986187	1.17244
0.404911	0.66343	0.676457	0.614597	0.587767
0.328068	0.626766	0.711821	0.439656	0.677476
0.365694	0.685085	0.731506	0.734062	0.554275
0.209521	0.733225	0.743581	0.574139	0.539326
0.332698	0.570048	0.572193	0.827711	0.80315
0.426394	0.707297	0.856001	0.468669	0.600307
0.419955	0.674797	0.688261	0.65285	0.790574
0.596968	0.94906	0.978396	0.811004	0.978321
0.141746	0.673562	0.699699	0.408039	0.276416
0.006814	0.757237	0.77956	0.966111	1.20151
0.335964	0.695124	0.762193	0.639316	0.589929
0.387444	0.7653	0.801821	0.500704	0.788963
0.446504	0.762132	0.811381	0.418279	0.654162
0.434648	0.741468	0.752628	0.502713	0.798526
0.148581	0.672473	0.704858	0.560712	0.501959
0.277885	0.609198	0.616689	0.546217	0.79285
0.515898	0.831448	0.864555	0.817576	0.939088
0.416478	0.740035	0.751916	0.646529	0.634302
0.501331	0.723644	0.912352	0.664617	0.932843
0.509036	0.766422	0.813723	0.695787	0.409979
0.36322	0.820578	0.87571	0.550339	0.769686
0.688797	0.995259	1.2001	1.05158	1.1038
0.373922	0.570876	0.598719	0.772511	0.719875
0.258045	0.571873	0.590185	0.43599	0.757185
0.444747	0.836667	0.865644	0.617447	0.765807
0.352394	0.601087	0.744963	0.467075	0.798242
0.270181	0.440491	0.451138	0.701208	0.758097
0.424914	0.632203	0.655896	0.802093	0.885733
0.333816	0.652813	0.672604	0.761188	0.622868
0.36288	0.650145	0.677462	0.519109	1.06936
0.350648	0.766728	0.784538	0.493327	0.689857
0.33146	0.59195	0.592193	0.246425	0.656854
0.498328	0.691082	0.695515	0.670327	0.692942
0.332628	0.672686	0.736844	0.410552	0.453782
0.390672	0.64435	0.807268	0.671196	0.957688
0.385112	0.615702	0.709124	0.514998	0.803567
0.307521	0.507279	0.612201	0.3906	0.714119
0.325847	0.530277	0.543263	0.588691	0.867143
0.459303	0.80205	0.844266	0.641618	0.931584

0.66135	1.05432	1.07706	1.21428	1.15376
0.482948	0.842765	0.871552	0.773302	0.67933
0.364897	0.717676	0.756158	0.548112	0.561783
0.484895	0.854303	0.864004	0.507001	0.899705
0.429329	0.667064	0.691676	0.576933	0.850577
0.260947	0.510207	0.511659	0.53802	0.824869
0.279606	0.66808	0.682125	0.700325	0.77815
0.403179	0.681168	0.689208	0.71922	0.38974
0.512501	0.662874	0.720201	0.536046	0.713025
0.415197	0.736997	0.741772	0.577972	-0.374256
0.333392	0.627931	0.630918	0.791598	0.863134
0.370412	0.547631	0.738726	0.734292	0.847836
0.406804	0.661783	0.737961	0.692022	0.675154
0.674979	0.940238	0.952932	0.932725	1.01269
0.226288	0.68564	0.700794	0.579375	0.899412
0.539662	0.703172	0.704272	0.68393	0.900177
0.234783	0.732198	0.823551	0.776755	0.868794
0.512886	1.00707	1.02314	0.851831	0.877798
0.452955	0.870559	0.890593	0.620693	0.966283
0.432067	0.565152	0.565793	0.454465	0.496717
Average				
0.379345	0.707228;	0.745586	0.643496	0.725219

Erdös- Rényi

10 links				
$a(G_1)$-$a(G_0)$	**$a(G_2)$-$a(G_0)$**	**$a(G_3)$-$a(G_0)$**	**$a(G_4)$-$a(G_0)$**	**$a(G_5)$-$a(G_0)$**
0.74322	1.44617	2.41556	1.64014	0.756607
0.632658	1.15485	2.63479	0.964394	0.647555
0.523073	0.827558	0.889231	0.302037	0.04614
0.024263	2.07852	2.28364	0.161135	0.073486
0.562433	1.03564	1.50986	1.45427	0.613371
0.026473	1.54064	2.08531	1.80491	0.069692
0.143053	0.539225	0.568784	1.85548	0.077326
0.63846	1.26344	1.29549	0.737134	0.017066
0.085776	0.612099	0.748349	0.158682	0.19984
0.789841	3.28169	3.90731	0.144644	0.037601
0.042027	2.57695	2.70258	0.89603	0.513978
0.03955	2.27012	2.29921	1.83928	0.031867
0.059559	1.16324	1.29956	0.279049	0.100871
0.039694	0.290909	1.31495	0.751346	0.175788
0.897775	1.70289	4.56953	3.71462	0.180782
0.024651	1.49038	1.55484	1.66827	0.038889
0.03618	0.512237	1.98924	1.78173	0.072314
0.047795	0.734796	0.851847	0.01411	0.147589
0.144709	1.01383	1.44268	2.01318	0.475386
0.358858	0.428979	0.6026	1.37376	0.073938
0.03991	0.425913	0.614362	1.90725	0.730043

0.033382	0.959951	1.00435	0.007653	0.886336
0.74917	2.46172	2.56858	1.07177	0.035909
0.594648	0.970402	0.972124	0.774525	0.866164
0.00378	3.41741	3.4468	1.87786	0.080327
0.03737	0.526386	0.891768	1.22707	0.140489
0.153289	0.572605	0.621936	1.77478	0.451228
0.096304	0.777998	1.21068	0.572509	0.160257
0.029397	4.53078	4.55451	5.14495	0.696999
0.008217	4.07598	4.11387	0.410278	0.059211
0.117713	0.529855	0.704865	0.810819	0.231132
0.037734	0.240103	0.959852	0.552764	0.373637
0.082814	0.717482	1.10806	1.29071	0.129079
0.892472	3.14045	3.29533	2.59097	0.648228
0.128731	0.917119	1.2438	0.546028	0.065122
0.265547	0.521968	0.741495	0.330964	0.095688
0.828373	2.28936	2.32239	0.056964	0.79048
0.071196	0.80211	1.08338	0.937157	0.665848
0.096816	0.569581	1.83044	0.075072	0.021667
0.029477	0.967123	1.19149	1.22802	0.079554
0.107201	1.16682	1.17573	0.019849	0.164058
0.153674	0.658157	0.810922	0.68401	0.031567
0.014314	0.959785	1.15824	0.713594	0.057854
0.088403	0.606015	1.84843	1.73915	0.057327
0.011865	2.15237	2.6081	0.410086	0.085417
0.457256	1.06091	1.68222	2.07306	0.064995
0.055542	1.2858	2.10611	0.005481	0.274504
0.035942	1.90596	2.02539	2.45154	0.10881
0.103342	0.392755	0.655473	0.622576	0.072511
0.061348	2.24989	2.35345	1.47957	0.089402
0.131369	1.46153	1.64661	2.0411	0.062074
0.039284	0.271828	3.01681	1.294	0.12848
0.046161	0.23577	0.405767	0.225064	0.038295
0.839021	4.40948	4.45844	0.730094	0.942435
0.056435	2.08581	2.15659	0.080033	0.321331
0.076879	0.721447	0.974114	1.80898	0.103021
0.732522	1.85826	1.89584	0.272245	0.065
0.058054	2.2617	2.90066	2.95427	0.545646
0.819728	1.2983	1.96502	0.528441	0.112329
0.028956	3.09705	3.24039	3.59748	0.040312
0.117236	0.605514	0.93888	1.06109	0.827959
0.015457	2.1978	2.35374	0.517288	0.601895
0.01412	3.56739	3.68808	2.68354	0.126652
0.010884	1.83202	1.86672	2.59119	0.166796
0.257696	0.704247	0.706791	3.88198	0.936399
0.368884	1.25402	1.47385	2.04714	1.0349
0.029023	2.81899	2.82191	3.02245	0.167276
0.023912	0.889599	1.80492	0.756437	0.233044
0.543923	1.35327	1.45259	0.165167	1.08217
0.053959	0.254336	0.728169	0.829481	0.091587
0.012782	2.1206	3.04375	1.99118	0.270315

0.083868	0.270842	0.383714	1.70944	0.125171
0.029373	0.673763	1.05457	1.07888	0.34206
0.551159	1.00428	1.33262	0.538836	0.754372
0.464453	0.944528	1.47399	0.206493	0.117212
0.499544	0.983502	1.00305	0.394187	0.058143
0.002751	1.72575	3.3058	1.58501	0.115339
0.438324	0.784404	1.00132	0.035756	0.093669
0.038979	1.86164	2.00156	1.07265	0.102727
0.021842	3.78005	3.90293	2.56776	0.086409
0.710252	1.79658	3.40751	0.780349	0.106386
0.021771	3.87201	4.15177	0.915123	0.401272
0.012201	4.25287	4.7638	3.64477	0.099209
0.016725	6.04645	6.07841	6.85529	0.873179
0.057867	2.08488	2.268	2.02311	0.236603
0.895183	5.09601	5.29871	3.09144	0.069497
0.114479	0.819806	1.35602	0.90431	0.131197
0.412498	1.16957	1.22075	1.14714	0.090633
0.021887	4.05864	4.08577	3.74455	0.367797
0.113263	0.647945	1.08239	0.757857	0.024491
0.093433	1.77142	2.47533	0.859368	0.405945
0.029473	0.531943	0.567679	1.16269	0.053019
0.016741	3.19964	3.69694	2.50379	0.909094
0.039539	0.754242	1.35828	1.20936	0.360455
0.614899	0.772201	2.80356	0.406425	0.0561
0.095631	0.619059	0.676657	0.749318	0.442388
0.536143	1.37935	1.51233	0.519273	0.924506
0.029685	2.97219	4.13837	2.50028	0.026313
0.071361	2.70642	2.91738	2.22435	0.048653
0.051648	0.587948	0.600342	1.5248	0.044751
				0.055946
Average				
0.216065	1.60284	2.01329	0.412197	0.27712

50 links				
a(G_1)-a(G_0)	**a(G_2)-a(G_0)**	**a(G_3)-a(G_0)**	**a(G_4)-a(G_0)**	**a(G_5)-a(G_0)**
0.952909	2.63703	2.69396	1.02649	1.85934
1.92645	2.50211	2.76864	2.56819	1.55007
0.398319	1.31519	2.27684	0.979012	1.12848
2.24534	3.83396	3.9306	3.87546	1.1639
1.43501	3.06778	3.06896	0.528472	2.7052
1.49518	2.31162	2.35113	0.290189	1.05231
0.947371	2.77861	2.92601	0.294178	2.74114
0.698661	1.86603	1.88833	0.141808	1.81045
1.22983	0.456187	1.23993	0.518454	1.04269
3.78604	4.70569	4.86094	3.52161	1.54835
0.768402	1.65563	1.93579	1.04302	1.74409
1.32593	2.11864	3.40209	0.132555	1.19402
1.71943	2.11985	3.25678	0.697002	1.34379
1.97611	2.15755	2.20315	2.79776	1.1428
2.26607	4.96835	4.99166	3.26658	1.10697

0.848088	1.93861	1.94082	0.71901	1.18948
1.32942	2.97097	3.05745	0.359175	2.65492
1.83044	4.67541	4.77803	3.15429	1.45317
0.501778	0.787857	0.800979	0.544875	1.53594
0.958278	0.190631	1.50111	0.694701	0.582247
0.819874	1.76155	1.7727	2.04408	1.7892
2.81129	3.6632	3.79511	2.03658	1.9326
2.33038	4.1312	4.15539	0.646042	2.27619
0.758204	0.138658	0.103246	1.22347	1.56127
2.67325	4.98613	5.50347	2.97282	1.28332
1.82959	2.32791	2.69098	0.973404	1.1254
1.75363	0.210971	0.092976	1.86811	1.09977
0.307461	1.6968	2.62	0.097144	1.14694
4.25369	5.16435	6.09944	4.89099	0.997436
6.32335	7.13733	7.27916	1.70603	2.10881
0.539133	0.679009	0.942343	2.61395	1.58444
3.63407	3.81892	5.58474	1.32295	1.5791
1.71717	2.82946	3.07928	0.693854	2.37989
0.774202	3.15191	3.31007	2.59738	1.95721
0.312828	0.711438	1.07179	0.503398	1.10156
1.82544	2.97679	3.41588	0.158152	1.78056
1.61828	3.517	4.17567	4.04266	1.0726
0.645818	1.80576	2.52716	0.619687	1.5531
3.9719	5.06735	5.06852	2.74251	0.931185
2.36572	3.04895	4.86131	2.9287	2.48328
2.75631	0.573409	0.411644	1.93461	2.43658
0.493311	2.41212	2.43495	0.367667	1.18909
1.19074	0.927245	0.959855	0.036171	1.10662
0.476148	2.50545	2.54204	1.56531	0.933469
0.048283	0.700707	2.85335	0.342203	1.79246
0.861666	2.9033	2.91864	2.75182	2.17573
0.380989	1.92497	2.25223	3.01226	0.408645
2.84653	4.01	4.07182	4.84032	1.75519
0.112248	1.13638	1.32167	0.445421	2.36886
0.771798	2.71569	2.77096	2.05938	2.89112
0.675858	2.90653	2.93177	1.57369	2.65743
1.90977	2.73369	2.74122	1.88813	2.00151
0.915117	0.691213	0.822361	1.41974	1.24066
3.98981	5.43068	5.44293	1.05599	1.92963
0.27912	2.13856	2.17222	0.17185	1.89795
1.18127	4.28863	4.2896	1.70632	2.38884
1.55935	1.822	1.83096	2.38739	1.84807
0.267233	2.0959	3.1577	0.599143	1.25706
2.21809	3.86352	4.00806	2.46295	1.06732
2.62363	4.86761	5.0677	4.93042	1.8522
1.64372	0.891797	1.98646	3.54976	2.68734
2.41823	3.41551	3.45607	1.06213	1.06632
2.63402	5.1529	5.16031	1.86428	1.27374
2.94793	4.12595	4.13565	1.61044	1.55649
0.436045	1.50646	1.61851	0.685638	1.4592

0.938503	2.1779	2.37149	0.250809	1.12085
3.01743	3.15595	5.01248	0.831104	1.66538
1.24511	2.41627	3.1604	0.173521	2.64256
1.30149	4.04734	4.05237	2.65103	1.10965
0.982819	2.08657	2.14252	0.985005	1.85201
2.30333	2.83032	4.63175	1.75534	1.02997
0.849387	0.384495	1.61334	0.280102	1.90361
1.90069	1.31753	1.31996	0.360638	1.10857
1.22824	1.37479	1.3801	0.421538	1.19652
1.25474	2.26702	4.67455	0.22341	2.7074
1.48911	0.455347	0.359842	0.626305	1.74016
0.256001	1.605	3.77705	4.36106	1.72556
0.058943	0.473728	0.616752	1.26025	1.63861
2.23935	3.07672	3.97611	2.56424	1.16707
5.17771	6.50845	6.61514	5.46625	0.320671
3.78186	4.10756	4.8529	0.282688	1.08082
2.64041	3.60539	3.73417	3.7778	1.02729
2.47188	3.21971	3.22001	2.09246	2.74964
8.21923	9.15623	9.41851	6.27919	2.77599
3.02346	4.08121	4.08224	2.16099	1.8117
0.873157	5.06988	5.10635	4.81842	1.18103
0.851552	0.086627	2.29816	2.77246	0.492826
0.089926	2.95204	2.95326	1.1226	1.7979
5.27128	6.71499	6.89282	2.54031	1.50345
2.31585	0.458238	0.42981	0.826937	1.24394
0.313243	0.740983	0.975619	0.242598	1.16952
0.592229	1.01401	1.21799	0.00646	0.66322
4.10037	4.35907	5.35517	5.19345	1.87303
2.00948	3.49431	3.57751	1.20187	2.1185
0.8888	1.6422	1.72903	1.39818	1.20922
0.445431	0.889838	1.28599	2.83503	1.1217
1.91187	2.36114	2.36356	0.578979	1.88582
3.27436	4.28006	4.33561	0.913645	1.38522
3.3111	3.84513	3.90029	2.64707	1.59684
0.358606	1.59639	1.7156	1.05633	1.56162
				1.0797
Average				
1.1269	2.66018	3.07735	1.14406	1.56593