Comparing measures of network robustness

Research Paper Business Analytics

Ellen van der Meer

Comparing measures of network robustness

Research Paper Business Analytics

Ellen van der Meer

July 6, 2012

Supervisors:
Prof. dr. ir. R.E. Kooij
W. Ellens MSc

Prof. dr. R.D. van der Mei

TNO
Technical Sciences
Performance of Networks & Systems
Brassersplein 2
2612 CT Delft

VU Amsterdam
Faculty of Sciences
Business Analytics
De Boelelaan 1081a
1081 HV Amsterdam

Preface

As a student of the master Business Analytics (formerly Business Mathematics and Informatics) at VU University Amsterdam, one of my tasks is writing a research paper. This research paper has been written at TNO, a large organisation for applied scientific research in the Netherlands. I have been working in the Technical Science Expertise Group, within the Performance of Networks & Systems department (PoNS) under the supervision of Rob van der Mei (VU), Rob Kooij (TNO) and Wendy Ellens (TNO), whose survey of robustness measures [2] served as a basis for this research paper. Within PoNS, a lot of applied research on networks, their graphs and performance is conducted. With a lot of (business) mathematicians and econometrists, similarities with some of the Business Analytics master components are easy to find.

Within my master I chose the optional course 'Performance analysis of communication networks', taught by Rob van der Mei. In this course we apply probability theory and scheduling models on communication networks. In a guest lecture Rob Kooij gave during this course, he made the remark that the same theory is applicable to metro networks. This specific context interested me, and when he mentioned he always had room for interns, I made a mental note that this might be interesting for my research paper 'course' at the end of the year.

And indeed, I ended up doing a project for Rob Kooij, not directly linked to the metro networks but nevertheless a very interesting one. The subject of the project was to compare different mathematical robustness measures. By calculating the robustness of theoretical and real network graphs, we wanted to show that not all of the measures were pointing in the same direction. It turned out working on this subject was fun and I enjoyed my time at TNO very much. I would like to thank Rob Kooij for creating such an explicit assignment and PoNS for making me feel welcome from the very first moment. I would like to thank all three of my supervisors Rob, Rob and Wendy for their input, support and trust.

Summary

Over the years, the number of application areas where graphs have been used to study networks has grown. Examples of this are power grids, telephone networks, motorways, the railway system and digital virus spread, as well as the more traditional biological virus spread. Mathematical graph theory can be used to evaluate the networks and possibly help to improve them, or to protect humans and systems against viruses. Part of this is done by looking at the robustness of networks. This means we try to find out how well a network will behave when it is under attack or when it endures failures.

Different robustness measures have been developed over the years, but little is known about the relation between all these measures. In this paper we want to gain a fundamental insight into these relations by checking the consistency of robustness measures. The main question therefore is:

For each combination of robustness measures, is it possible to find a pair of graphs such that the robustness measures point in opposite directions?

We call these examples contradictions or counterexamples: there is no consistency among the measures.

To adress this question, first, we discuss the eleven robustness measures that were used. For all measures, we answer three questions: *What does the measure mean?*, *What is its formula?* and *How should we interpret the outcomes?* (i.e. the outcome range and what value indicates high robustness).

Then we have a look at sixteen simple graphs. For each graph we calculate the robustness values and we discuss with what other graphs it is comparable. This way we can easily make the comparison and search for incompatibleness among the measures. Next we have a look at some large networks: three hypothetical testbed networks, the network of US motorways and three IP backbone networks. Again we calculate the robustness values and discuss to what extent the measures are inconsistent with each other.

The main results of our work are the following:

- It turned out to be quite easy to find graph pairs that show incompatibility between the robustness measures.
- It was possible to find examples for all measure combinations, this is something that was not clear from the beginning.
- For a lot of graph pairs and measures similarities were visible as well.
- For large networks we found the same results: inconsistencies as well as trends are visible.

Contents

1 Introduction **1**
- 1.1 Relevance . . . 1
- 1.2 Aim of this paper . . . 1
- 1.3 Outline . . . 1

2 Definitions and methods **3**
- 2.1 Basic notations and definitions . . . 3
- 2.2 Mathematical background . . . 3
- 2.3 Software . . . 4

3 Robustness measures **5**
- 3.1 Edge connectivity . . . 5
- 3.2 Average distance . . . 5
- 3.3 Efficiency . . . 5
- 3.4 Clustering coefficient . . . 6
- 3.5 Algebraic connectivity . . . 6
- 3.6 Number of spanning trees . . . 6
- 3.7 Effective graph resistance . . . 6
- 3.8 Natural connectivity . . . 7
- 3.9 Percolation limit . . . 7
- 3.10 Resilience factor . . . 8
- 3.11 Graph diversity . . . 8

4 Comparing graphs **10**
- 4.1 Graphs . . . 10
- 4.2 Calculated measures . . . 11
- 4.3 Comparison . . . 11
- 4.4 Similarities . . . 12

5 Real life networks **14**
- 5.1 Real networks . . . 14
- 5.2 Network robustness . . . 15

6 Conclusion and further research **17**

Bibliography **19**

A Measure abbreviations **21**

1 Introduction

1.1 Relevance

In day to day life, we make use of a lot of networks, although most people will not be aware of this. Think for example of public transportation networks, power grids, social networks, motorways, the internet and telephone networks. These networks exist in the real world and thus might possibly suffer from failures. These can be due to human errors, technical failures or attacks. Think about the fire on the Vodafone network in the Netherlands on the fourth of April of this year. A lot of people who subscribe to the Vodafone network did not have access to telephone or internet connections for multiple days. This fire is an example of a very big failure or attack and the Vodafone network did not appear to be very robust: Vodafone has no systematic alternative solutions installed for when a significant part of their network breaks down. Another way of looking at this is evaluating a network in terms of the possibilities it has for virus spreading. In this case, you could say a network is robust if there is little possibility for (biological or digital) viruses to spread.

Over time, a lot of robustness measures for networks have been developed, but little is known about how they relate to each other. This would be interesting to know, because when it turns out that certain topologies are typically linked to high robustness values, this can be used to change other topologies and decrease failures.

All measures (or *metrics*) that are discussed here, are topological measures. This indicates that the measures only take the shape of the network into account, not the meaning or background of the network. For example in the case of a study of train networks, different paths that are used are not taken into account, nor do we look at the fact that not all trains (intercity trains, local trains) visit all stations (here the vertices) although they are using the same track (edges).

1.2 Aim of this paper

In this paper we will want to gain a fundamental insight into the relationship between a series of robustness measures that has been developed. We do this by comparing eleven of these measures. The aim of this paper therefore is to answer the question:

For each combination of robustness measures, is it possible to find a pair of graphs such that the robustness measures point in opposite directions?

We call these examples contradictions or counterexamples: there is no consistency among the measures. To avoid condusion, we emphasize that is not our goal to determine what the best robustness measure is, nor to find the optimal topology.

To be able to answer the main question, we start with gaining an insight into the eleven measures. For all measures, we answer three questions: *What does the measure mean?*, *What is its formula?* and *How should we interpret the outcomes?* (i.e. the outcome range and what value indicates high robustness).

Next, we look at how different graphs are represented by the different measures. We discuss the differences between the graphs and what graphs can be compared to what extend. (E.g. we prefer the graphs in a comparison pair to be of the same size.) What follows is a comparison of all measures and the search for contradictions.

Finally, we will calculate the robustnesses some large networks and compare these values.

1.3 Outline

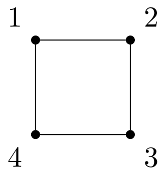
The remainder of this paper is organized as follows. In section 2, we start with explaining some basic notations, mathematical concepts and the software that was used. A reader who is very familiar with the field of graph robustness might want to skip this part. Next, we arrive at the two main parts of the paper. In section 3, the robustness measures that were used are discussed. For all measures, we tried to answer three questions: what does the measure mean, what is its formula and how should we interpret the outcomes (i.e. the outcome range and what value indicates high robustness). A list of abbreviations that are used to refer to the measures in the tables through the paper, can be found in Appendix A. The next main section, section 4, contains an overview of 16 theoretical graphs. Here we calculate their robustnesses and we make the comparisons between the measures. Next is section 5 with some real (IP) networks. After explaining the networks and their origins, we conduct the robustness calculations for these networks as well. In section 6 we conclude by summarizing the different outcomes and discussing them.

2 Definitions and methods

2.1 Basic notations and definitions

Because networks are studied in different fields, different names and notations are being used. In this paper we will stick to the (mathematical) graph theoretical notation. When we look at a graph G, we denote this as $G = (V, E)$ where V are the *vertices*, with $|V| = n$ and E are the *edges*, with $|E| = m$. (i, j) indicates the edge (connection) between vertices i and j. In other fields such as network theory, you might encounter the words nodes (for vertices) and links (indicating edges). In this paper we focus on simple, *undirected*, *unweighted*, *connected*, finite and deterministic graphs. This means that if there is a connection (i, j), the connection (j, i) exists as well (undirected). All connections have equal length, weight or costs (unweighted), and there is no vertex in the graph that is not connected to any other vertex (connected).

The adjacency matrix of a graph G is denoted by $A(G) = (a_{ij})_{N\times N}$, where $a_{ij} = x$ if there exists an edge with weight x from vertex i to j, otherwise $a_{ij} = 0$. As explained before, $a_{ij} = 1$ for unweighted graphs, $a_{ij} = a_{ji}$ for undirected graphs. If the i^{th} row and column both only contain zero, this is seen as a disconnected vertex i. Figure 1 shows cycle graph H and its adjacency matrix $A(H)$.



$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1. Graph H and its adjacency matrix

2.2 Mathematical background

In multiple measures, the concept of *vertex degrees* is being used. This degree δ_i of a vertex i expresses the number of connections the vertex has. When making use of adjacency matrices, this can be seen as the sum of the elements in a row (or column, these values will be equal for undirected graphs).

Next to vertex degrees, a notion that is often used is *eigenvalues*. The eigenvalues for a matrix B of size n can be found by solving $Bx = \lambda x$ where x is the eigenvector of the matrix and λ an eigenvalue. A well-known way to derive λ (in the case the eigenvector is not known) is solving the characteristic polynomial, which is known to be the outcome of the characteristic equation $det(B - \lambda I) = 0$ where I is the identity matrix with ones on the diagonal, zeroes elsewhere and with a size n similar to B. The characteristic polynomial will have the form $(y_1 - \lambda)(y_2 - \lambda)\cdots(y_n - \lambda) = 0$.

Note that for some measures the eigenvalues of the adjacency matrix are calculated, where other measures make use of eigenvalues belonging to the *Laplacian*. This Laplacian is a matrix constructed as follows, $L = D - A$ where A is the adjacency matrix and D is the degree matrix of the graph which has vertex degrees on its diagonal and zeroes elsewhere. This construction of the Laplacian of graph H is shown in Figure 2. In the example in the last paragraph, B can be a Laplacian or adjacency matrix.

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

Figure 2. The Laplacian constructed from the degree matrix and the adjacency matrix of graph H

2.3 Software

With the development of graph and network theory, together with the different robustness measures, some software has been developed. We tried and used different programs like Cytoscape, newGRAPH and Gephi, which all aim on clear visualisation of graphs. The programs can be used for calculations as well, for example Cytoscape can return the clustering coefficient and average distance. Still, it turned out that Matlab is the most useful program because new functions (like new robustness measures) can easily be added. This is especially true when making use of “The MatlabBGL library” [4] in Matlab. This library already includes different measures and methods to loop through graphs. It can return all kinds of connectivity and makes it easy to spot shortest paths, for example.

In contrast with the other programs mentioned, in Matlab a visual input of a graph can not directly be used for calculations. Instead, we make use of the adjacency matrices of the networks. As mentioned before, if the i^{th} row and column both only contain zero, this is seen as a disconnected vertex i. For programming (as we do in Matlab), it is an important insight because turning all inputs for a certain vertex to 0 does not remove the vertex, it only disconnects it.

We performed all computations in Matlab, except for the calculations of the resilience factor. Still, the other programs where useful, especially to validate the Matlab functions we wrote ourselves. This validation part is extremily important, especially when you rely heavily on the outcomes of certain calculations. Minor errors in (Matlab) scripts can have major effects, e.g. with a small mistype values can be multiplied. Apart from using the other software for this objective, validation was done by comparing our values with values we found in the different papers that describe or discuss the robustness measures.

The resilience factor was not implemented completely correctly in Matlab, and therefore was calculated by hand. The algorithm to find the total graph diversity was developed by the authors of [7], and we were happy to be able to make use of this.

3 Robustness measures

In this section we explore the different robustness measures we are going to compare. We discuss four typical classical graph measures, three spectral graph measures and four recently proposed measures.

In sections 4 and 5, we refer to the measures with abbreviations. These abbreviations are listed in Appendix A.

3.1 Edge connectivity

The first measure we discuss is the concept of *edge connectivity*, which is less general than the classical *connectivity* measure. The latter distinguishes only between connected graphs ($\kappa = 1$) and unconnected graphs ($\kappa = 0$), which lacks at least one connecting edge between a pair of vertices.

Next to this general connectivity and the frequently used measure edge connectivity, *vertex connectivity* is often used as well and comparable with edge connectivity. The edge (vertex) connectivity κ_e (κ_v) represents the minimal number of edges (vertices) that has to be removed to disconnect the graph. So the edge (vertex) connectivity of a graph depends on the least connected part of the graph. The measured value is integer because it represents a number of edges (vertices) and $\kappa_e \geq 1$. (If $\kappa_e = 0$ the graph is disconnected.) The larger κ_e, the harder it is to disconnect the graph and thus the graph is considered to be more robust. These three statements also hold for κ_v. [2]

3.2 Average distance

The *distance* measure d_{ij} is defined as the length (number of edges) of the shortest path between vertices i and j. d_{max} is the maximum over the distances, also known as the *diameter*. The *average distance* over all pairs is denoted by $\bar{d}$ and is equal to $\frac{2}{n(n-1)}$ times the sum of the lengths of the shortest paths (the Wiener index) [2]:

$$\bar{d} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij}$$

Values can take any number larger than or equal to 1, where $\bar{d} = 1$ means all vertices are directly connected with each other. Therefore it holds that the shorter the path, and thus the smaller the average distance, the robuster it is.

3.3 Efficiency

Similar to the distance, we have the *efficiency* measure E, which is the averaged sum of the reciprocal (multiplicative inverse) of the distances. Thus, instead of d_{ij} we now have $\frac{1}{d_{ij}}$ in the formula:

$$E = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{d_{ij}}$$

Because the formula consists of the reciprocal of the distance, the range of outcomes is the reciprocal as well. Therefore values are in the interval $(0, 1]$, so here as well $E = 1$ means all vertices are directly connected with each other. Because we take the inverse, it now holds that the greater the value, the greater its robustness. [2]

In a paper by Latora and Marchiori, this type of efficiency is called *average efficiency* or *global efficiency*. They also define *local efficiency*, which is the average of the efficiency of all subgraphs of a network. More on this topic can be found in [5].

3.4 Clustering coefficient

The *clustering coefficient* captures the presence of triangles, and compares the number of triangles to the number of connected triples. This coefficient returns the portion of vertex pairs j, k sharing a neighbour i where j and k are also neighbours themselves. The clustering coefficient c_i of a vertex i is defined as the number of edges e among neighbours of i, divided by the total possible number of edges among its neighbours: $\delta_i(\delta_i - 1)/2$. Here δ_i is the degree (number of neighbours) of a vertex i.

The overall clustering coefficient of a graph is the average over the clustering coefficients of the vertices, which can be expressed as the sum of the ij-th elements aij of the adjacency matrix A [2]:

$$C = \frac{1}{n} \sum_{i \in V; \delta_i > 1} c_i = \frac{1}{n} \sum_{i \in V; \delta_i > 1} \frac{2}{\delta_i(\delta_i - 1)} e_i =$$

$$\frac{1}{n} \sum_{i \in V; \delta_i > 1} \frac{2}{\delta_i(\delta_i - 1)} \sum_{j=1}^{n} \sum_{k=1}^{n} a_{ij} a_{jk} a_{ki} = \frac{1}{n} \sum_{i \in V; \delta_i > 1} \frac{2}{\delta_i(\delta_i - 1)} (A^3)_{ii}$$

A high clustering coefficient indicates high robustness, because a lot of triangles mean a lot of alternative paths in case of failures on a vertex or edge. The clustering coefficients range between 0 and 1, where 1 indicates all vertices are interconnected so all possible triangles exist.

3.5 Algebraic connectivity

The *algebraic connectivity* is the second smallest eigenvalue of the Laplacian L, which has been explained in section 2.2. The eigenvalues of this matrix L can be ordered from small to large $(0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n)$, now the algebraic connectivity is equal to λ_2. [2]

If $\lambda_2 = 0$ the graph is disconnected, so a higher value means the graph is more robust. Most values are between zero and one, but can be equal to the number of vertices when all vertices are interconnected.

3.6 Number of spanning trees

A *spanning tree* is a subgraph containing $n-1$ edges, all n vertices and no cycles. This definition is not exclusive and therefore a graph can have multiple different spanning trees. As a robustness measure, we count all possible spanning trees that exist for a graph. The number of spanning trees is normally integer, and can grow quite big. It has been proven that the number of spanning trees ξ can be written as a function of the unweighted Laplacian eigenvalues [2]:

$$\xi = \frac{1}{n} \prod_{i=2}^{n} \lambda_i$$

3.7 Effective graph resistance

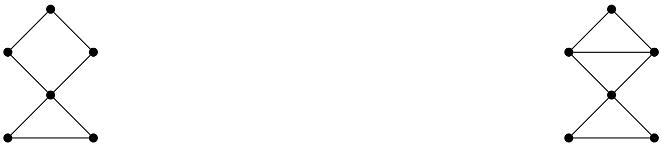
When a graph is seen as an electrical circuit, you can see an edge (i, j) corresponding to a resistor or $r_{ij} = 1$ Ohm. The *effective resistance* R_{ij} between two vertices i, j of a network (when a voltage is connected across them) can be calculated by series and parallel manipulations. The effective graph resistance is the sum of the effective resistances over all pairs of vertices. It is proven this can also be written as a function of non-zero Laplacian eigenvalues:

$$R = \sum_{1 \leq i = j \leq n} R_{ij} = n \sum_{i=2}^{n} \frac{1}{\lambda_i}$$

A small value of the effective graph resistance indicates a robust network. The values vary significantly and can grow up to up to several hundred thousands. [3]

3.8 Natural connectivity

In their paper in 2008, Wu et al. [9] propose *natural connectivity* as a robustness measure. Natural connectivity is based on the number of closed walks in a graph, that can be related to the sum of eigenvalues. A walk of length k is a path over the vertices and edges of the graph, starting at v_0, passing $k-1$ vertices and k edges to end in v_k. If $v_0 = v_k$ this path is called a closed walk. A closed walk may contain repeated vertices, so the length can be infinite. A closed walk of length $k = 2$ will only reach one other vertex and pass the same edge twice, a closed walk of length $k = 3$ is a triangle. So closed walks are directly related to subgraphs and can serve as a measure for networks. Wu et al. introduce a measure based on the sum of numbers of closed walks. They argue that for some graphs (e.g. M and N in Figure 3) with identical edge connectivity and algebraic connectivity, a distinction can be made by their proposed natural connectivity.



(a) Graph M

(b) Graph N

Figure 3. Graph M and N with identical edge connectivity of 2 and algebraic connectivity 0.7369 but different natural connectivity.

Let S be the weighted sum of numbers of closed walks, $S = \sum_{k=0}^{\infty}(\frac{n_k}{k!})$ where n_k is the number of closed walks of length k. Using matrix theory, we find $n_k = \sum_{i=1}^{n} \lambda_i^k$ where λ_i is the i^{th} largest eigenvalue of the associated adjacency matrix. Now we can write $S = \sum_{i=1}^{n} e^{\lambda_i}$ and we can calculate the scaled 'average eigenvalue': the natural connectivity (or natural eigenvalue). For the full derivation, see [9].

$$\bar{\lambda} = ln\left(\frac{S}{n}\right) = ln\left(\frac{\sum_{i=1}^{n} e^{\lambda_i}}{n}\right)$$

λ_i was the i^{th} largest eigenvalue, so in accordance with the notation used in this paper it should be denoted as λ_{n+1-i}. Nevertheless, because we sum e^{λ_i} over all i and the order does not change the outcome, this can be neglected.

In our calculations, $\bar{\lambda} \in [0, 2]$, where a higher value indicates a more robust graph.

3.9 Percolation limit

The *percolation limit* (or *percolation threshold*) returns the critical fraction of nodes that need to be removed before the network disintegrates (disconnects). The percolation limit is used to study the robustness of the internet. Here p is the fraction of the nodes (vertices) and their connections (edges) of a graph/network that is (randomly) removed. Here, we calculate the treshold p_c which means that if $p > p_c$ the network desintegrates into smaller, disconnected parts.

According to [1], $1 - p_c = (\kappa_0 - 1)^{-1}$ where $\kappa_0 = \frac{\langle k_0^2 \rangle}{\langle k_0 \rangle}$, for the original graph. In an example with n vertices v_1, v_2, ..., v_n with corresponding vertex degrees δ_1, δ_2, ..., δ_n, these values k_0 and k_0^2 are defined in such a way that:

$$\langle k_0 \rangle = \frac{\delta_1 + \delta_2 + ... + \delta_n}{n}$$

$$\langle k_0^2 \rangle = \frac{\delta_1^2 + \delta_2^2 + ... + \delta_n^2}{n}$$

So we can calculate

$$p_c = 1 - \frac{1}{\frac{\langle \kappa_0^2 \rangle}{\langle \kappa_0 \rangle} - 1}$$

making use of the original adjacency graph, we do not have to simulate random removals. Of course, κ_0 is not allowed to equal 1 but this will not happen for connected graphs with more than two vertices. Because p_c is a fraction, values are expected to be in the $[0, 1]$ range, but it can also become negative. Therefore we suggest to let $p_c = \max 0, p_c$. Nevertheless, in the remainder of this paper, we will make use of the limit as defined by Cohen et al. For futher interpretation we have to know that a higher percolation limit indicates the fraction of vertices that can be removed without problems is higher, which means the network is more robust.

3.10 Resilience factor

When we consider robustness (or similarly *resilience*), we look at the capability of a network to remain in service when vertices or edges fail. The *resilience factor* expresses the fraction of subgraphs where 1 up to $n-2$ vertices are removed and where the remainings are still connected. A graph with $n-1$ vertices removed consists only of one vertex, and is therefore not of interest. For a graph with n vertices, removing i vertices can be done in $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ ways. Now $k(i)$ denotes the number of connected subgraphs divided by this total of $\binom{n}{i}$ for each set of subgraphs with i vertices removed. Now the resilience factor R_F can be constructed as the average of these fractions:

$$R_F = \frac{\sum_{i=1}^{n-2} k(i)}{(n-2)}$$

In [8] i ranges from 2 up to $n-1$, $k(i)$ here represents the fraction of connected subgraphs where $i-1$ vertices are removed. The k we use here does not correspond with the k used for calculations of the percolation limit.

Values are between zero and one, as we are working with fractions. A higher R_F is better because this indicates the graph will stay connected in more cases of vertex removal than a graph with lower R_F.

3.11 Graph diversity

Another way of checking the ability of a graph to remain connected, is by calculating the *path diversity.* For two vertices in a graph, the different paths that are possible between both vertices are determined. For every path, we look at the number of vertices it shares with the shortest path. A diversity of 1 expresses the fact that the paths do not share a vertex. This means that in case of failure, the combination of the paths guarantees robustness. So this is more robust than a diversity of $\frac{2}{3}$ which indicates an overlap of $\frac{1}{3}$. Over all paths the *effective path diversity* (EPD) is calculated, and the average of all EPD's is denoted as the *total path diversity* (TPD), which is the measure we included in our comparison.

To calculate the path diversity, let the shortest path between a pair of vertices (i, j) be P_0. The diversity function $D(x)$ with respect to P_0 is defined as $D(P_k)$ for any other path P_k between (i, j). So

$$D(P_k) = 1 - \frac{|P_k \cap P_0|}{|P_0|}$$

where $|P_0|$ is the total number of vertices and edges of the shortest path for (i, j), $|P_k \cap P_0|$ is the combined number of vertices and edges that P_0 and P_k share.

Now the effective path diversity (EPD) is $EPD = 1 - e^{-\lambda k_{i,j}}$ where $k_{i,j}$ is a measure of the added diversity, defined as:

$$k_{i,j} = \sum_{i=1}^{k} D_{min}(P_i)$$

The total graph diversity (TGD) can now be defined as the average of the EPD values of all node pairs within that graph, $TGD = \sum_{i=1}^{n} \sum_{j=i+1}^{n}$ for undirected graphs, for directed graphs the second summation should be $\sum_{j \neq i}^{n}$.

Total graph diversity was calculated with the original Matlab files of Rohrer and Sterbenz, who proposed these measures in [7]. Here you can choose k, the number of alternative paths that is used for the calculations. As the outcomes converge when k increases, we let Matlab increase this value until the outcomes differed less than 0.00001.

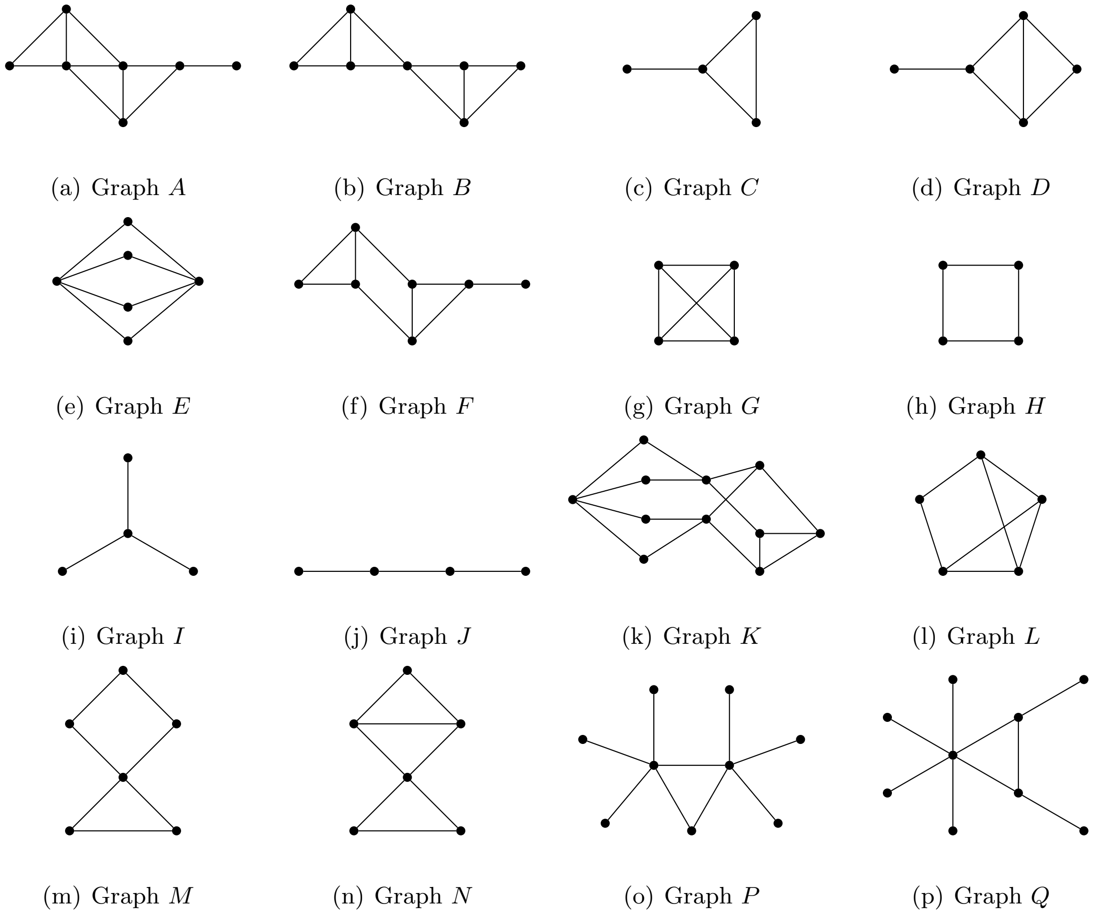
All values range between $[0, 1]$. As mentioned in [7], a star topology has a TGD of 0 (graph I), a ring has a TGD of 0.6 given $\lambda = 1$. As a ring is generally considered as more robust than a star, and given the path diversity explanation, a bigger TGD indicates greater robustness.

4 Comparing graphs

In this section, we are going to make a comparison between the different robustness measures. In order to do this, we calculate the robustness of different graphs. We will explain the graphs we used in section 1. Then we calculate the robustness of theoretical graphs in different ways in section 2. Next we compare the outcomes and list some of the contradictions in section 3. Section 4 describes some cases where we did not find a contradiction.

4.1 Graphs

We considered 16 simple, undirected, unweighted graphs that we found in the literature. Some are more or less similar and thus hopefully comparable, others are very different. The graphs are shown in Figure 4.



(a) Graph A (b) Graph B (c) Graph C (d) Graph D

(e) Graph E (f) Graph F (g) Graph G (h) Graph H

(i) Graph I (j) Graph J (k) Graph K (l) Graph L

(m) Graph M (n) Graph N (o) Graph P (p) Graph Q

Figure 4. The graphs that were used for the comparison of the measures.

We divided the graphs into several groups, depending on their numbers of vertices and edges. These groups were subsequently divided into three categories, containing the same kind of relations. In sections 4.3 and 4.4, we will refer to these categories to value the comparisons we made.

category 1 similar number of vertices and similar number of edges: $\{A, B\}$, $\{C, H\}$, $\{E, N\}$, $\{I, J\}$, $\{P, Q\}$

category 2 only similar number of vertices: $\{A, B, F\}$, $\{C, G, H, I, J\}$, $\{D, L\}$, $\{E, M, N\}$

category 3 no similar size, this can be a combination between any two graphs

4.2 Calculated measures

Except for the resilience factor (RF) which has been calculated by hand, implementation of the measures in Matlab made it easy to calculate the robustness measures for all graphs. Graphs K, P and Q were too big to calculate the RF by hand, these spots are therefore left empty. The results are displayed in Tables 1 and 2, with the values of a certain measure in each row, and the graph that is being evaluated in each column. As discussed in section 2 for the individual measures, for nearly all values it holds that the bigger the value, the more robust the graph is. Only average distance (AD) and effective graph resistance (EGR) should be interpreted the other way around i.e. a smaller value indicates higher robustness.

	A	B	C	D	E	F	G	H
n	*7*	*7*	*4*	*5*	*6*	*7*	*4*	*4*
m	*10*	*10*	*4*	*6*	*8*	*9*	*6*	*4*
EC	1	2	1	1	2	1	3	2
AD	1.7619	1.8095	1.3333	1.5000	1.4667	1.8095	1	1.3333
EF	0.7024	0.6944	0.8333	0.7833	0.7667	0.6786	1	0.8333
CC	0.5238	0.7143	0.5833	0.5333	0	0.3810	1	0
AC	0.6338	0.5858	1	0.8299	2	0.5858	4	2
NST	55	64	3	8	32	30	16	4
EGR	21.6727	21.5000	6.3333	10.2500	11.5000	23.7333	3	5
NC	1.5140	1.4457	0.9857	1.2345	1.2517	1.2307	1.6672	0.8676
PL	0.5455	0.5000	0.2000	0.4000	0.5000	0.4375	0.5000	0
RF	0.5371	0.6229	0.7083	0.6667	0.7000	0.5276	1	0.8333
TGD	0.6455	0.6083	0.4105	0.5043	0.6533	0.5850	0.8647	0.6321

Table 1. The robustness values for graphs A up to H.

	I	J	K	L	M	N	P	Q
n	*4*	*4*	*11*	*5*	*6*	*6*	*9*	*9*
m	*3*	*3*	*19*	*7*	*7*	*8*	*9*	*9*
EC	1	1	2	2	2	2	1	1
AD	1.5000	1.6667	1.9091	1.3000	1.6667	1.6000	2	2
EF	0.7500	0.7222	0.6318	0.8500	0.7111	0.7444	0.5833	0.5833
CC	0	0	0.2909	0.4000	0.3611	0.7778	0.1333	0.0815
AC	1	0.5858	0.8440	2	0.7639	0.7639	0.4586	0.5300
NST	1	1	12625	24	12	24	3	3
EGR	9	10	41.8337	6.4167	16	14.5	64	64
NC	0.6716	0.6465	1.6937	1.3986	1.0879	1.3508	1.0315	1.0332
PL	0	-0.5000	0.6200	0.4615	0.3636	0.4667	0.5714	0.5714
RF	0.6250	0.5000	-	0.8667	0.4375	0.4542	-	-
TGD	0	0	0.8781	0.7925	0.5825	0.5980	0.1925	0.1925

Table 2. The robustness values for graphs I up to Q.

4.3 Comparison

In Table 3 graph combinations that result in a counterexample are shown. On both sides, the 11 different measures are displayed. In a certain box, the names of two graphs that result in a contradiction are displayed. Input “A:B” means the combination of graphs A and B results in a ‘counterexample’ for the robustness measures in that corresponding row and column.

As we could see, the graphs we calculated are not all very similar. They have different structures and different numbers of vertices and edges. When we compare the values, we have a preference for using pairs of graphs of similar size, that is similar number of vertices and if possible also similar number of edges. Therefore we preferred the comparison of the pairs of category 1. A counterexample with one of these pairs is marked bold: **A:B**. Next, we looked at category 2 but it turned out that all contradictions in this category were already covered by the first one. Next in line, we have the couple $\{C, D\}$. Although they do not share the same number of vertices nor edges, you can say D is an extension of C. This couple is emphasized by italic typewriting: *C:D*. If we did not find a counterexample in one of these categories, we denoted an example of non-similar graphs (category 3). These are not marked (e.g. D:E). We did not look at the difference in the values: a contradiction with $\mathrm{NC}_A = 1.5140$ and $\mathrm{NC}_J = 0.6465$ is not preferred over $\mathrm{NC}_P = 1.0332$ and $\mathrm{NC}_Q = 1.0315$.

	EC	AD	EF	CC	AC	NST	EGR	NC	PL	RF	TGD
EC	X	**A:B**	**A:B**	**C:H**	**A:B**	F:G	B:C	**A:B**	**A:B**	C:E	**A:B**
AD	X	X	D:E	**A:B**	J:K	**A:B**	**A:B**	**E:N**	*C:D*	**A:B**	*C:D*
EF	X	X	X	**A:B**	D:E	**A:B**	**A:B**	**E:N**	*C:D*	**A:B**	*C:D*
CC	X	X	X	X	**A:B**	**C:H**	**C:H**	**A:B**	**A:B**	**C:H**	**A:B**
AC	X	X	X	X	X	**A:B**	**A:B**	**C:H**	**C:H**	**A:B**	*C:D*
NST	X	X	X	X	X	X	*C:D*	**A:B**	**A:B**	*C:D*	**A:B**
EGR	X	X	X	X	X	X	X	**A:B**	**A:B**	D:F	**A:B**
NC	X	X	X	X	X	X	X	X	**E:N**	**A:B**	**C:H**
PL	X	X	X	X	X	X	X	X	X	**A:B**	**C:H**
RF	X	X	X	X	X	X	X	X	X	X	**A:B**
TGD	X	X	X	X	X	X	X	X	X	X	X

Table 3. The graph combinations that result in contradictions.

With $\{A, B\}$ we could already find 30 of the 55 contradictions. 41 of the couples are in category 1 and thus share the same number of vertices and edges. By adding $\{C, D\}$ we could fill in another 7 spots. For the last couples, we used category 3. For a lot of combinations, more than one pair of graphs was possible although only one of them is shown here. Especially the clustering coefficient (CC) had a lot of ‘preferred’ contradictions.

For a lot of combinations of graphs and measures, we did not find a contradiction but only a difference. Hereby we mean that for some graph couples, one measure did make a difference between the graphs and the other measure treated them exactly the same. We did not need such examples to fill the comparison table. Instead, it turned out we did not need all sixteen graphs to get all comparisons, ten would have been enough. For example, we could have left out H, I, L, M, P and Q.

The average distance (AD) and effective graph resistance (EGR) were only contradictive in one example (D:E), and here both values were still very close to each other. This could be expected from the literature: for some cases it is proven that $R_{ij} = d_{ij}$ [3].

4.4 Similarities

For the graphs in categories 1 and 2, we expected similarities in the robustnesses because of their comparable shapes. For some of these nicely comparable graphs where there was little variation between the values calculated, this hypothesis was confirmed. Some of the observations are reported here.

B:F is not an interesting comparison: for all measures B is considered to be more robust than F or they are equal (AD, AC).

A:F gives similar results. They have the same edge connectivity (EC) and in all other cases A is more robust.

D:L is not an interesting combination either. L is more robust except for the CC were D is larger.

I:J are equal in four cases (EC, CC, NST, TGD), in all other cases I is robuster than J. G:H:I:J (looking at C:H was enough)

M:N is a combination where N has one edge more than M. Therefore you would expect N to be robuster, and indeed all measures show this. Wu et al [9] argued that natural connectivity was a good alternative because sometimes other measures canot distinguish differences between graphs and they gave the example of graphs M and N. With the natural connectivity measure the two can be distinguished, but as it turns out there are a lot of other measures that can help distinguishing between the two. Actually, in our 'selection' *only* EC and AC yield exactly the same values.

P:Q is an interesting couple because they have exactly the same size: the same number of vertices and the same number of edges. This equality is also visible in their robustnesses: nearly all of them are equal. The comparison results in only two contradictions: CC/AC and CC/NC.

5 Real life networks

In this section we look at real networks and how they are evaluated by the different robustness measures. Now that all Matlab functions are implemented, we want to test some real networks. We loaded some networks from ResiliNets [6]. The site can return unweighted and weighted adjacency files. We used the unweighted ones so our approach and results will be similar to what we did earlier with the theoretical graphs. Matlab has a nice import function, and after converting the matrices into sparse matrices we could calculate them with the Matlab functions.

5.1 Real networks

Before we start evaluating different networks, we have a look at what these networks are. The networks available on the website have between 5 and 573 cities (vertices) and between 0 and 540 links (edges). Most of the networks are IP backbone networks, large networks of devices in worldwide computer networks.

Two of the networks we consider are *GpENI L1* and *GpENI L2*. The abbreviation GpENI stands for ‘Great plains Environment for Network Innovation’, which is an international programmable network testbed. This testbed is constructed to enable research on internet architecture. The L1 network is a subset of the much larger L2 network. GpENI L1 has the shape of a star graph like graph I, but now with a total of 5 vertices instead of 4.

Similarly, *Coronet L1* is a network from the Darpa Coronet Program. The L1 network is a global hypothetical fiber-optic backbone network developed for use in research. Of the total of 100 cities and 136 links, 76 cities and 100 links are to be found in the US. The other cities are in Europe and Asia.

US motorways corresponds with the Interstate Highway System in the USA, a network of over 47000 miles (nearly 76000 km) of public road length. A visualisation of the US Motorways as a network can be seen in Figure 5, which is created with a Web-based topology map viewer on the website [6]. In this section we use the abbreviation *USmw* for this network.

AT&T is a large U.S. telecommunication company and the *AT&T L1* network we consider here is an IP backbone network constructed out of a subset of their data. Similarly, the network *Sprint L1* corresponds to the global voice, data and internet services provider Sprint and *Tiscali L3* which is linked to the Italian company Tiscali. Most of the datapoints in the Tiscali network can be found in Europe. On ResiliNets more networks of AT&T and Sprint can be found, as well as networks from other internet providers.

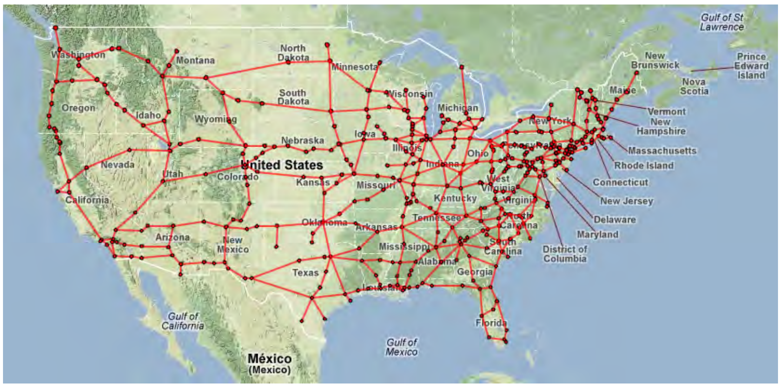


Figure 5. The US motorway network in KU-TopView.

5.2 Network robustness

For the seven networks that were introduced in the previous section, we calculated some robustness measures. The results can be found in Table 4. None of the networks was one to one comparable with regards to the number of vertices and edges. The resilient factor (RF) of the GpENI L1 was calculated by hand, this was possible because this is a relatively small network. For the other networks RF was not calculated because of their size. This also resulted in a problem for calculation of the TGD. The program starts with calculating all minimal paths and then searches alternative paths. For a graph of 200 vertices, this is quite a job and it seems Matlab runs into problems. Unfortunately, we therefore have not been able to calculate values for this metric.

	Coronet L1	GpENI L1	Tiscali L3	GpENI L2	USmw	AT&T L1	Sprint L1
n	*100*	*5*	*51*	*51*	*400*	*361*	*236*
m	*136*	*4*	*129*	*61*	*540*	*466*	*311*
EC	2	1	1	1	1	1	1
AD	6.6741	1.6000	2.4298	4.6965	13.3395	13.5739	14.7751
EF	0.2045	0.7000	0.4693	0.2803	0.1074	0.1064	0.1017
CC	0	0	0.3776	0.1847	0.0526	0.0487	0.0314
AC	0.0508	1	0.5255	0.0533	0.0059	0.0061	0.0053
NST	1.5873e26	1	1.9348e22	5.5638e05	2.8195e92	8.5742e73	1.1732e41
EGR	1.0857e04	16	1.3256e03	4.5582e03	3.2213e05	2.8257e05	2.0127e05
NC	1.1339	0.7443	5.6718	1.2212	1.2151	1.1499	1.0075
PL	0.4925	0.3333	0.8974	0.6187	0.5400	0.5136	0.3890
RF	-	0.6000	-	-	-	-	-
TGD	-	0	-	-	-	-	-

Table 4. The robustness values for some 'real' networks.

The edge connectivity does not make any difference for six of the graphs, although it would help if you could look at the number of ways a graph can be disconnected by removing a number of edges equal to the edge connectivity. This is captured by yet another robustness measure, the reliability polynomial. We did not include this in our overview because it has a very large correspondence with the edge connectivity and therefore you would not expect to find any contradictions. Still, for these networks visible in Table 4 it is interesting to notice that this value can distinguish them and can show that the networks are indeed very different. The network disconnection possibilities of GpENI L1, GpENI L2, USmw, AT&T L1 and Sprint L1 are respectively 4, 14, 34, 40 and 21, where a value of 4 means that there are 4 vertices that are connected to the network with only one edge. For the remaining part of the network, the edge connectivity is higher, and therefore that part is more robust. The value of 40 therefore tends to express that the AT&T L1 network is less robust than the Sprint L1, because there are more possibilities to disconnect it by only removing one edge.

When neglecting Coronet L1, the networks are now sorted in such an order that the average distance is increasing. This indicates that, out of the remaining six, GpENI L1 is the most robust network. Neither the edge connectivity nor the efficiency measure contradicts this. Here we have to keep in mind that for the average distance a small value indicates robustness, where for the efficiency a larger value means greater robustness. Nevertheless, the other measures do not support the order that would be expected from the first couple of measure outcomes, so the counterexamples we were looking for earlier can be found here as well. A representation of this ordering, with 1 for the most robust network and 7 for the least robust, can be found in Table 5. For cases with similar outcomes (EC, CC), fewer numbers are used.

	Coronet L1	GpENI L1	Tiscali L3	GpENI L2	USmw	AT&T L1	Sprint L1
EC	1	2	2	2	2	2	2
AD	4	1	2	3	5	6	7
EF	4	1	2	3	5	6	7
CC	6	6	1	2	3	4	5
AC	4	1	2	3	6	5	7
NST	4	7	5	6	1	2	3
EGR	4	1	2	3	7	6	5
NC	4	7	1	2	3	5	6
PL	5	7	1	2	3	4	6

Table 5. An ordering of the robustness values for some 'real' networks, 1 meaning most robust, 7 indicating least robustness.

The number of spanning trees depends on the size of the graph, and therefore any comparison between unequal matrices seems unfair. Still, it is striking to see that most networks are repeatedly awarded the same 'position'. GpENI L1 gets the opposing positions 1 and 7 repeatedly, but this might be influenced by the size of this small network. GpENI L2 is in nearly all cases ranged as second or third, and Tiscali L3 also receives good scores. Coronet L1, AT&T L1 and Sprint L1 on the other hand do not seem to do very well. For these networks, it seems that the different robustness measures score them more or less according to a trend.

6 Conclusion and further research

To gain insight in the relation between different robustness measures that have been developed over the years, we compared eleven of those measures in this paper. The main question we wanted to answer is: *For each combination of robustness measures, is it possible to find a pair of graphs such that the robustness measures point in opposite directions?* At the beginning we were not sure whether it would be possible to find examples for all measure combinations and we had no idea how hard it would be to find nice graphs. It turned out that it was not that difficult to find inconsistencies between measures.

To address the main question, we started with gaining insight into the eleven robustness measures and we chose some theoretical graphs where we calculated the measures for. With these values we made a comparison between robustness measures and graph couples. For seven large, real world networks we did the same thing; we calculated their robustness values and compared the results (even though the networks we used were not easily comparable in size).

After carrying out these comparisons, we can come to four main results of our work:

- It turned out to be quite easy to find graph pairs that show incompatibility between the robustness measures.
- It was possible to find examples for all measure combinations, this is something that was not clear from the beginning.
- For a lot of graph pairs and measures similarities were visible as well.
- For large networks we found the same results: inconsistencies as well as trends are visible.

It turned out it is important to mention for all values what the range of the outcomes is and what values indicate high robustness. These two factors can easily lead to misunderstandings. It would therefore be easier if there is a way to rescale all measures in such a way that all measures return values between $[0, 1]$, where 1 is high robustness and 0 is not robust (i.e. a disconnected graph). This can be something of interest for further research.

We have been able to fill 41 of the 55 measure combinations with contradictions with graph pairs of the same size. This means the two graphs had the same number of vertices and edges. For 14 measure combinations, we were not able to find such combinations. It would be interesting for future work to focus on finding graphs of the same size for these 14 positions as well.

References

[1] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85(21), 2000.

[2] W. Ellens and R.E. Kooij. Graph measures and network robustness. preprint, 2011.

[3] W. Ellens, F.M. Spieksma, P. Van Mieghem, A. Jamakovic, and R.E. Kooij. Effective graph resistance. *Linear algebra and its applications*, 435:2491–2506, 2011.

[4] D. Gleich. Matlab BGL v4.0, a Matlab graph library, 2008. `http://www.stanford.edu/~dgleich/programs/matlab_bgl/`.

[5] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19), 2001.

[6] J.P. Rohrer. Resilinets topology map viewer, 2011. `http://www.ittc.ku.edu/resilinets/maps`.

[7] J.P. Rohrer and J.P.G. Sterbenz. Predicting topology survivability using path diversity. RNDM 2011, 2011.

[8] R.M. Salles and D.A. Marino Jr. Strategies and metric for resilience in computer networks. *The Computer Journal Advance Acces*, October 19, 2011.

[9] J. Wu, Y.-J. Tan, H.-Z. Deng, Y. Li, B. Liu, and X. Lv. Spectral measure of robustness in complex networks, 2008. arXiv: 0802.2564.

A Measure abbreviations

In sections 4 and 5, some abbreviations have been used to refer to the robustness measures. Here we list the abbreviations used.

n = number of vertices

m = number of edges

EC = edge connectivity

AD = average distance

EF = efficiency

CC = clustering coefficient

AC = algebraic connectivity

NST = number of spanning trees

EGR = effective graph resistance

NC = natural connectivity

PL = percolation limit

RF = resilient factor

TGD = total graph diversity