Effective Resistance and DYNER Measure examined

Sande, J.E.A. van de

Mathematisch Instituut

Universiteit Leiden

Effective Resistance and DYNER Measure examined

by

Joëlle van de Sande

Supervisor: **F. Spieksma**

Submission date: 10-08-2020

Bachelorscriptie

Contents

Introduction

In this thesis, we will discuss two measures that evaluate network robustness in some kind of way. Network robustness, the ability to withstand perturbations and failures, is an essential concept for many fields. Successes of many processes depend on having a course as smooth as possible, yet also being able to overcome certain problems that might occur. Each network is different and therefore every network could ask for another measure to evaluate its robustness. One might care about the infection rate of a disease and views a disease with a higher infection rate as more robust, but another might want to look at extraction from a network. Each example could require a different measure.

Due to the needed diversity, it will not come as a surprise to learn that there are more than a few robustness measures. Over time, more and more measures follow. To be able to preserve depth within this thesis, we have chosen to restrict ourselves to two measures. These two measures are the Total Effective Resistance and the DYNER measure, from which the latter has been defined within the last decade.

We will mainly focus on processing a part of the known theory and trying out a few ideas of ours. However, we also add to the existing knowledge by providing easy accessible code and examples, next to demarcating the two measures a little more. The thesis is divided into two chapters and also includes an appendix with the fabricated code. The first chapter will cover the theory we needed to understand what we wanted to research. Logically, the second chapter adds our input to the subjects and contains interesting examples. The code is provided separately, might one want to use it for further research or understanding, together with some commentation.

Chapter 1

Theory

This section is meant as an overview of some theory on the subjects we want to cover. With this we hope to provide knowledge to our readers so that they can get familiar with the subject, before we add our examples, code and other additions.

Within this chapter, we will refrain from repeating the conditions we require for every graph. When we talk about a graph G or $G(V,E)$ without further ado, we mean a simple, undirected, connected graph $G(V,E)$. Here, V is the finite set of nodes and E is the set of edges. The chapter is divided into two parts, one covering the theory on the Dyner Measure, and the other going through important theory on the Effective Resistance.

1.1 The DYNER Measure

As also introduced by Y.Singer in his article [4], the Dynamic Network Robustness measure (short: DYNER measure) enables analysis of network robustness by considering backup in the network. During the last decade and even before, researchers have become more and more interested in graph representations with node and/or edge extraction. For instance, examining a terrorist attack on a train station, causing it to be 'out of order'. Travellers planning to travel via this station are forced to take a different route, which might make their journey last longer. Another contemporary example is a hacker attacking the internet, causing certain websites to be offline. When browsing the internet, you might find that the easy access to your desired information, will take you multiple redirections instead of the usual one via a site that is now offline.

Researchers want to know how well their networks are resistant to node extraction, which makes a measure that measures precisely that type of resistance quite interesting. The key elements are the so-called back-up vertices, that yield the network its resistance. So far, not much research has been done on this measure. Its creator Y.Singer claims the measure to be the only one that considers backup in the network in instances of node deletion. However, the proof that goes with Singer's article [4] supporting this claim, contains mistakes and is hard to follow. We will go more into detail in the next chapter.

In order to fully tap into the calculation of and the rest of the theory about this measure, we will first need to get our definition of the named backup vertices straight. Next, with this definition fresh in our mind, we will provide the means to calculate this measure. After that we include a simple statement about both weighted and directed graphs, since we want to be as thorough as possible because of the fact that we do cover those types of graphs with the Effective Resistance measure.

1.1.1 Backup Vertex of order k

Y.Singer in his article [4], proposed a definition of a backup vertex of order k, but we would like to alter the definition a bit in order to be more precise. The definition as being given in Singers article poses a fundamental problem for (possibly) proving monotone increase of the Dyner Measure with respect to the rise in backup. The higher the outcome of the measure, the more robust the measured network is (i.e: the better it is). A robustness measure is created fueled by the idea that certain change makes the network more robust. In case of the DYNER measure, the so-called 'change' indicates the rise in backup. So when a change (e.g: added vertice or edge, weight change) to the network gives rise in backup, the DYNER Measure should monotonically increase. Otherwise it is not a good robustness measure. Also, another reason for changing the original definition, the definition as being given by Y.Singer can be interpreted in multiple ways (does a node count as its own backup if it is a neighbour of node v?). Therefore we define a backup vertex of order k in the following way:

Definition 1 - Backup Vertex

In a graph $G(V, E)$, a node $v \in V$ with a non-empty set of neighbours N_v, has a backup vertex of order $k \geq 1$ with $k \leq |N_v|$, if there is a node $u \in V\backslash_{\{v\}}$ such that u is connected to k nodes in $N_v\backslash_u$ (neighbours of v, u itself excluded) in $G\backslash_{\{v\}}$ (the graph with node v extracted).

1.1.2 Calculation

The next definition is equal in its implementation to Y.Singer's [4], however formulated in a way that we prefer. When comparing the two fractions below, the role of backup vertices is not immediately clear, since the definition does not directly include them. The answer whether they make or break the measure can only be answered when Singer's proof -monotone increase when there is rise in backup- is either verified or contradicted.

Definition 2 - DYNER Measure

The DYNER measure of graph $G(V,E)$ can be calculated with the following equation:

$$\Gamma(G) := \left[\sum_{w \in V} \sum_{u \in V \setminus w} \left(\frac{1}{\sum\limits_{v \in V \setminus \{w,u\}} d(u,v)} - \frac{1}{\sum\limits_{v \in V \setminus \{w,u\}} d_w(u,v)} \right) \right]^{-1},$$

where $d(u,v)$ denotes the shortest path length between nodes u and v and $d_w(u,v)$ denotes the shortest path length between nodes u and v, when node w is extracted from the graph.

For these shortest path lengths the following properties hold:

- $d(u,v) \geq 0$, for $u, v \in V$
- $d(u,u) = 0 \quad \forall u \in V$
- $d(u,v) \leq d(u,w) + d(w,v)$ for $u, v, w \in V$.

We also assume the following rules for the shortest path lengths:

- $d(u,v) = \infty \quad \iff$ there is no path between u and v, $u, v \in V$
- $\frac{a}{\infty} = 0 \quad$ for any constant a.

1.1.3 Expansion to both weighted and directed graphs

It is easy to see that allowing the edges of $G(V,E)$ to be weighted, does not cause any problems. That is, if there are no negative cycles, i.e. cycles whose edges sum up to a negative number. When there are no negative weights at all, one could use Dijkstra's algorithm for computation and in other cases with non-positive cycles, the Bellman-Ford algorithm will do the job. For a positive weight $r \in \mathbf{R}$, you can view the corresponding edge as r edges, where partial edges are allowed (with negative weight $-r$, use your imagination to view it as a shortage of r edges). All of the properties and rules denoted in the last section still hold and you can start calculating the DYNER Measure for your non-positive, cycle-lacking, directed, simple graphs using the equation in Definition 2.

Since both Dijkstra and Bellman-Ford allow graphs to be directed, allowing this type of graph poses no problems and Definition 2 is adequate.

1.2 Total Effective resistance

Another measure to analyse network robustness is the Total Effective Resistance, a term often used in electric circuit analysis, where it defines the resistance between two points in an electrical network. In graph theory, the Total Effective Resistance is used in the same way as the Dyner Measure, to get some kind of measurement of the graph construction for weighted, simple, connected graphs. Where the Dyner Measure works with distances to calculate the resistance value, the Total Effective Resistance has received multiple ways to be calculated. We will start by giving definitions of some of these ways to calculate the Effective Resistance. During which, we will focus on the application in graph theory and choose not to use electrical terms. After doing that, we will cover the knowledge we obtained about weighted and directed graphs in combination with the Effective Resistance.

It is useful to know that we think of the Total Effective Resistance, denoted by R^{tot}, as the sum of the Effective Resistances between each pair of nodes $\{i, j\}$, denoted by R_{ij}, which yields us

$$R^{tot} = \sum_{j>i} R_{ij}.$$

A lot of the theory in this part of the chapter has been obtained from W.Ellens' master thesis [1], where more information on the Effective Resistance can be found.

Where the DYNER Measure adores the backup vertices (or is said to do so), the Effective Resistance worships edge addition. In contradiction to the DYNER Measure, for the Total Effective Resistance monotonical increase was chosen. Intuitively one could say that less resistance means a better network. Adding an edge, yields a smaller Total Effective Resistance. Aside from that Ellens [1] also researched increasing weight for one edge, and proved that doing that also yields increase of the measure.

1.2.1 Calculation using Laplacian

As far as we know, this is the most recognized way to calculate the Effective Resistance. Before we dive into a formula for calculation of the Effective Resistance using the Laplacian, we give the common definition of the Laplacian.

Definition 3 - Laplacian Matrix

The Laplacian matrix L of graph $G(V, E)$ is a matrix of size $|V| \cdot |V|$ with

$$L_{ij} = \begin{cases} -1, & \text{if } (i, j) \in E, i \neq j \\ d(i), & \text{if } i = j \\ 0, & \text{else,} \end{cases}$$

where $d(i)$ denotes the degree of vertex i.

A known way to calculate the Effective Resistance using the Laplacian is done by using the Moore-Penrose pseudo-inverse L^+ of the Laplacian. In 1993, Klein and Randić [3] used that way to propose a formula for calculation and they also proved that its validity. Their result is given next.

Theorem 4 - Total Effective Resistance (eigenvalues)

The total Effective Resistance of graph $G(V, E)$ can be calculated with the following equation:

$$R^{tot} = n \sum_{i=2}^{n} \frac{1}{\mu_i},$$

where $\mu_2, ...\mu_n$ are the nonzero Laplacian eigenvalues (and μ_1 equals zero).

1.2.2 Calculation using The Matrix-Tree Theorem

Another way to calculate the Total Effective Resistance we are interested in, is by using the Matrix-Tree theorem. In order to get this equation, Cramer's Rule was used to rewrite an alternative version of Theorem 4 that uses the traditional way of finding a matrix inverse. The rewritten equation will be the following Theorem, since we need it for further understanding.

Theorem 5 - Total Effective Resistance (via Cramers Rule)

The total Effective Resistance of graph $G(V, E)$ can be calculated with the following equation:

$$R^{tot} = \sum_{j>i} \frac{\det L_{G,[i,j]}}{\det L_{G,[j]}},$$

where det *denotes the determinant of the matrix it precedes. $L_{G,[i,j]}$ and $L_{G,[j]}$ are the Laplacian matrix of graph G with columns and rows i and j omitted and the Laplacian matrice of graph G with column and row j omitted respectively.*

In order to (intuitively) understand this Theorem - and enter our own territory - we will substitute the terms in the equation. However, to do so we need the definitions of a Tree and a Spanning Tree, which we will provide below.

Definition 6 - Tree

Graph $G(V, E)$ is a Tree if it is connected and acyclic (i.e. there is exactly one path between any distinct pair of vertices.)

Definition 7 - Spanning Tree

A Spanning Tree of a connected graph $G(V, E)$ is a subgraph that has vertex set V and is a Tree.

From graph theory, we would like to recall that the sum of the weights of all of the spanning trees of graph $G(V, E)$ equals $\det L_{G,[j]}$ for any $j \in V$ (the Matrix-Tree Theorem).

What is even better is that a Corollary of the Matrix-Tree Theorem even provides us with the fact that total weight of the spanning trees of graph $G(V,E)$ containing a certain edge (a,b), regardless of whether it exists in E, equals $\det L_{G,[a,b]}$. Now we have all of the puzzle pieces needed to perform the substitution. The outcome is the following theorem that feeds our intuition as for the understanding of Theorem 5 (which, just like the next theorem, calculates the Total Effective Resistance).

Theorem 8 - Total Effective Resistance (via Matrix-Tree)

The total Effective Resistance of graph $G(V,E)$ can be calculated with the following equation:

$$R^{tot} = \sum_{j>i} \frac{\#\{\textit{spanning trees of graph G containing edge (i,j)**}\}}{\#\{\textit{spanning trees of graph G}\}},$$

*** edge added if not yet present in graph.*

1.2.3 Expansion to weighted graphs

The section before provided us with almost all of the right tools to expand the measure in order to allow calculation for graphs with weighted edges as well. For simplicity and correctness we assume that our now weighted graph $G(V,E)$ has at least 3 nodes. The last thing we need to do to adjust the calculation of the Total Effective Resistance for non-weighted graphs, is to adjust the definition of the Laplacian Matrix. As a source we have used the definition already present in Vos' thesis [5].

Definition 9 - Weighted Laplacian Matrix

The weighted Laplacian matrix L_w of graph $G(V,E)$ is a matrix of size $|V| \cdot |V|$ with

$$L_{w,ij} = \begin{cases} \sum_{k=1}^{|V|} w_{i,k}, & \textit{if } i = j \\ -w_{i,j}, & \textit{else,} \end{cases}$$

where $w_{i,j}$ denotes the weight of edge $(i,j) \in E$ and the weight 0 else.

With the weighted Laplacian and defining the resistance of any edge $(a,b) \in E$ equal to $\frac{1}{w_{i,j}}$, we can use Theorem 5 for our weighted graph as well. We have found an alternative (non-standard) proof to this in the masterthesis of V.Vos [5] that supports this claim. Just like the Laplacian, the Matrix-Tree Theorem also has a weighted option, which is that the sum of the weights of all corresponding spanning trees of graph $G(V,E)$ is equal to $\det L_{w,G,[j]}$ for any $j \in V$. Here $\det L_{w,G,[j]}$ is the determinant of the weighted Laplacian matrix L_w of graph G from which column and row j are omitted. So we can substitute the denominator of Theorem 5, using The Weighted Matrix-Tree Theorem, but in order to substitute the numerator of the fraction found in Theorem 5 for weighted graphs, we need slightly more work. Luckily this work has already been done conform V.Vos [5]. Therefore we will just write down the Theorem below.

Corollary 10 - Corollary of The Weighted Matrix-Tree Theorem

The Total Weight of the Spanning Trees of graph $G/_{a,b}$ equals $\det L_{G,[a,b]}$ Here $G/_{a,b}$ denotes the graph obtained by merging vertices $a, b \in V$ and removing the edges between them if present.

Straight from the Weighted Matrix-Tree Theorem and Corollary 10 we can immediately perform our substitution in the equation of Theorem 5 and yield the equation in the Theorem 11. This is all valid, since we can trade the Laplacian in Theorem 5 for its weighted version. For a proof for Theorem 11, we would like to refer to Vos' masterthesis [5].

Theorem 11 - Total Effective Resistance (weighted graphs)

The total Effective Resistance of weighted graph $G(V, E)$ can be calculated with the following equation:

$$R^{tot} = \sum_{j>i} \frac{\{\textit{sum of weights of all spanning trees of graph } G/_{i,j}\ \}}{\{\textit{sum of weights of all spanning trees of graph } G\}},$$

where $G/_{i,j}$ denotes the graph obtained by merging vertices $i, j \in V$ and removing the edges between them if present.

1.2.4 Expansion to directed graphs

In their two-part paper [6][7], G.Young, L.Scardovi and N.Leonard generalize the Effective Resistance in undirected graphs, with the result that it is now possible to do the calculation for directed graphs as well. We will start with the the reasoning behind the definition and we then will give some extra information. The definition is given beneath the following paragraph.

First and foremost it is important to know that 'connected' means there is a node k to which there exists a path in G from any given node i. So directed graphs not satisfying this requirement are excluded from this equational calculation. However, since for undirected graphs we require each pair of nodes to share a path between them, this restraint for directed graphs is less strict (a connected, directed graph is automatically connected when all edges are made undirected, but not necessarily the other way around). In the definition below there are some symbols and equations worth mentioning, might anyone try to find out more about them. In short: the $\overline{L}$ is called 'the Reduced Laplacian' and W is the solution to the 'Lyapunov equation' ($\overline{L}W + W\overline{L}^{\top} = I_{n-1}$, see Definition 12). The reason why it is required to have a connected, directed graph can be explained with that W is only invertible when the directed graph is connected, this information can be found in multiple scientific papers [6][2].

Definition 12 - Total Effective Resistance (directed graphs)

Let G be a directed, connected graph existing out of n nodes and having Laplacian matrix L. The total Effective Resistance of graph $G(V, E)$ can be calculated with the following equation:

$$R^{tot} = \sum_{j>i} \left[(e_n^{(i)} - e_n^{(j)})^\top X (e_n^{(i)} - e_n^{(j)}) \right] = \sum_{j>i} \left[x_{i,i} + x_{j,j} - 2x_{i,j} \right],$$

where

$X = 2Q^\top W Q,$

$\overline{L}W + W\overline{L}^\top = I_{n-1},$

$\overline{L} = QLQ^\top,$

$e_n^{(i)}$ is the i-th unit vector,
I_k is a k-dimensional identity matrix,
and Q is a matrix with rows that form an orthonormal basis for $1_n^\perp = span\{1_n\}^\perp$ (1_n is an n-dimensional all-ones vector).

Instead of calculating the Effective Resistance of a connected, directed graph one can also use the matrix Xc, to obtain its pseudo-inverse X^+. We will give the definition of X^+ below.

Definition 13 - The pseudoinverse

The pseudo-iverse of matrix X (symbol X^+) can be calculated using the following equation:

$$X^+ = \frac{1}{2} Q^\top W^{-1} Q,$$

where Q is a matrix with rows that form an orthonormal basis for $1_n^\perp = span\{1_n\}^\perp$ and W is the outcome of the Lyapunov equation.

Because X^+ is an $n \times n$ matrix that is symmetric, positive definite and has row and column sums zero, it can be interpreted as an undirected Laplacian matrix. Therefore, we then created an undirected graph with an equal total Effective Resistance as the initial connected, directed graph. K.Fitch [2] has given a couple of examples in her paper that we discuss in the reseach chapter of this thesis. She has also explored the relation between the Laplacian of connected, directed graphs and their accompanying undirected

Laplacian. As a side effect we show the equality of the outcome of the measure of the given pairs of graphs.

We could not find any research on what restrictions and changes to force onto the graph to get monotonic increase of the measure. The next chapter will include our research on the matter.

Chapter 2

Own Research

2.1 The DYNER Measure

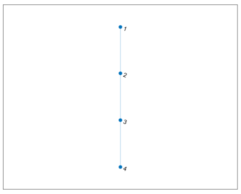
In order to fully understand the DYNER Measure, we started with calculating the measure for a few examples. After doing so, we then started to look into some interesting questions concerning this measure, such as whether there is a monotonical increase posing certain conditions on the graph. As being said in the last chapter, monotonical increase is an important property a robustness measure should possess (which makes the Total Effective Resistance the inverse of robustness measure) The DYNER measure for a connected, undirected graph is claimed to be monotonically increasing with respect to the rise in backup in the graph, by Y.Singer in his article [4]. He also wrote down a 'proof'. However, this 'proof' has mistakes in it and makes faulty assumptions along with that. Since we could not immediately prove nor deny the claim, we were interested whether adding edges in general would get the job done.

2.1.1 Some examples

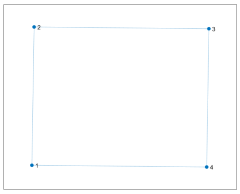
As been said, we tried out the DYNER Measure with a few common graph examples. Since doing it all by hand is very sensitive to errors we first wrote a Matlab program to compute the measure (Figure A.1). Below we will show the graphs we wish to measure (Figure 2.2) and by using our code we can also show the outcome of the DYNER measure for each graph (Figure 2.1).

G	P4	C4	K4	K3,3	C5	C5*
$\Gamma(G)$	0.6383	∞	∞	∞	3.0000	7.5000

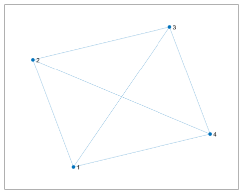
Figure 2.1: Outcome DYNER Measure for denoted graphs Figure 2.2



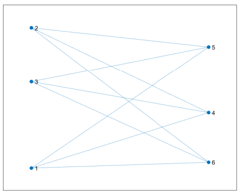
(a) Path graph P4



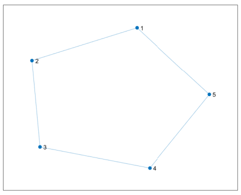
(b) Cycle graph C4



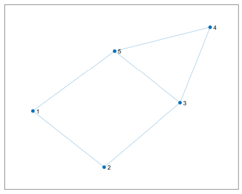
(c) Complete graph K4



(d) Complete graph K3,3



(e) Cycle graph C5



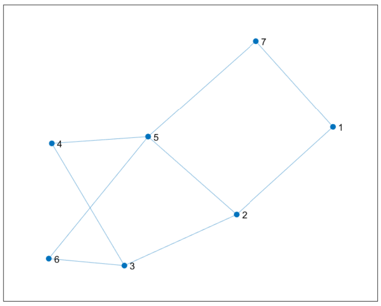
(f) C5*:= C5 with added edge

Figure 2.2: Some simple graphs

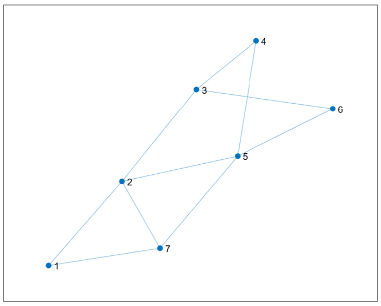
As one can see in the table in Figure 2.1 C4, K4 and K3,3 have an infinitely big DYNER Measure. This is no surprise because none of the shortest path lengths are affected by extracting a node. Since the same argument holds for any other complete graph, we hereby see that each will have an infinitely big DYNER Measure. Aside from that, we also see that adding one or more edges in some cases, P4,C4 and C5 to be precise, increases the DYNER Measure. This fueled our interest whether adding an edge would always increase the DYNER Measure.

2.1.2 No monotonical increase for adding an edge at random

We were wondering if adding an edge to a random connected undirected graph would monotonically increase the DYNER measure. If this would turn out to be true, we immediately can assume monotonical increase w.r.t. rise in backup to be true as well (since in a graph with the same nodes this can only possibly be done by adding certain edges). However, we found a counterexample proving adding an edge not to monotonically increase the DYNER Measure of the graph. This counterexample has seven nodes, with edgelist $E = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{5,7\},\{7,1\},\{6,3\},\{5,2\}\}$ (see Figure 2.3a). The edge we add is $\{7,2\}$ (see Figure 2.3b). In the table in Figure 2.4 we provided the outcome of our Matlab code for both graphs. The added edge acquires a decreased DYNER Measure, concluding our counterexample.



(a) Graph



(b) Graph with added edge

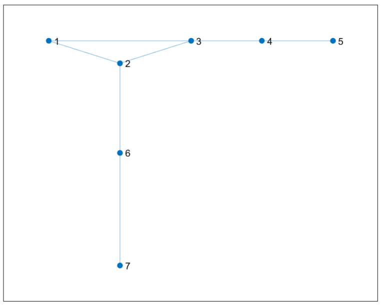
Figure 2.3: Graph representation counterexample graph

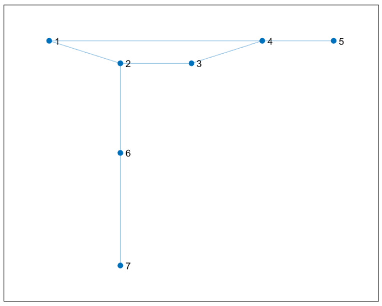
G	origional	with added edge
$\Gamma(G)$	10.4066	9.5104

Figure 2.4: Outcome DYNER Measure for counterexample

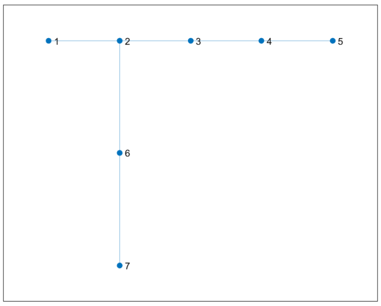
2.1.3 Paving the path towards monotonical non-decreasing

Looking at the counterexample in the last section, it is interesting to notice that the added edge, that causes the DYNER Measure to decrease, does not give any rise in back-up. We would like to show another example illustrating the importance of precisely which edge to add in order to get an increase or decrease. Also we will start to pave the way towards believing what Y.Singer [4] failed to correctly prove. The graph G_{ill} we will use as illustration, consisting out of 7 nodes and having edgelist $E = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{2,6\},\{6,7\}\}$, is shown in Figure 2.5c. To this graph we will add two edges and review the subsequent graphs separately. The edges we separately add are $\{1,3\}$ and $\{1,4\}$ (see Figure 2.5a and Figure 2.5b).





(a) Graph G_{ill} with added edge $\{1,3\}$ **(b)** Graph G_{ill} with added edge $\{1,4\}$



(c) Graph G_{ill}

Figure 2.5: Graph representations G_{ill} with and without additional edges

For each of the three graphs, we used our Matlab code to calculate the DYNER Measure, the results are shown in Figure 2.6. If one examines the graphs in Figure 2.5, in particular whether the additional edge causes rise in back-up or not, one can conclude that the edge $\{1,3\}$ creates back-up edges (1 and 3) for node 2. However the edge addition $\{1,3\}$ does not give a rise in the already existing back-up, while the additional edge $\{1,4\}$ does (4 was already a back-up node of order 1 of node 2 and after adding $\{1,4\}$, node 4 is now a back-up node of node 2 of order 2). Looking at the outcome of the DYNER Measure in Figure 2.6 for G_{ill} with added edge $\{1,3\}$, one notices a decrease, while the additional edge $\{1,4\}$ that gives rise in back-up, delivers an increase of the measure. Since our counterexample in the last section also added an edge not giving rise in back-up, we start to wonder whether the claim made by Y.Singer [4] is actually true. All of the edge additions we tried on graphs, that gave rise in back-up, always gave an equal or greater outcome of the DYNER Measure. All of the edge additions we found yielding a decrease of the measure, never gave any rise in back-up. Therefore the next thing one probably would like to do is try to come up with an actual proof, since, as we have mentioned before, the proof Y.Singer has provided is abbreviated and has some errors in it. Unfortunately, we have not been able to provide it until now and therefore encourage others to do so.

G	G_{ill}	G_{ill} + edge $\{1,3\}$	G_{ill} + edge $\{1,4\}$
$\Gamma(G)$	0.5012	0.4665	0.6037

Figure 2.6: Outcome DYNER Measure for G_{ill} + edge additions

2.2 The Effective Resistance

With this measure we will also start by trying it out for a couple of examples of undirected graphs. To be able to compare the studied measures, we decided on using the same examples. Yet, since they both take different characteristics of graphs into account, they turned out not to be so easy to compare. Since the extension to directed graphs is more complicated for the Total Effective Resistance than the DYNER Measure, we were interested in some directed graphs for calculation as well.

2.2.1 Some examples (undirected)

Again we would like to stay free of calculation error and decided to write a Matlab program to do the computational work. In the program we used the calculation method in Theorem 4. This program we have included in the appendix section of this thesis (Figure A.2) as a reference. Since we aim to compare both measures, we consider the same graphs (Figure 2.2), and we extended the corresponding table with the outcome of the Effective Resistance measure. See below this paragraph in Figure 2.8.

G	P4	C4	K4	K3,3	C5	C5*
$\Gamma(G)$	0.6383	∞	∞	∞	3.0000	7.5000
R^{tot}	10	5	3	9	10	8.1818

Figure 2.7: Outcome Effective Resistance next to outcome DYNER Measure for denoted graphs Figure 2.2

The outcome is very much as expected. First, notice that both measures have different outcomes. This is to be expected, since they do not measure the same thing. Whereas the DYNER Measure measures the extraction of nodes, the Effective Resistance sums the resistances between pairs of nodes. The Effective Resistance did not increase in our examples when edges were added, but W.Ellens [1] already concluded from a theorem described by Klein and Randić [3] that the Effective Resistance is non-increasing when edges are added. This, as has been said in the last chapter. Therefore the outcome is not very surprising.

2.2.2 Some examples (directed)

Having explained the generalization that makes it possible to calculate the total Effective Resistance for connected, directed graphs in the previous chapter, of course we would like to provide a couple of examples. This calculation is by far the worst to do by hand, therefore we wrote an extensive piece of code that one can find in the Appendix section of this thesis (Figure A.3) that we have used to calculate the total Effective Resistance for the upcoming examples. In her article K.Fitch [2] provided a table of examples together with the undirected graphs yielding the same total Effective resistance, see Chapter 1, and since we can at the same time check our code using our previous code on the undirected accompanying graphs, we decided on using those examples here in this section. Figure 2.9 shows visual representations of the graphs used as examples and below we will show the table with the outcomes of our Matlab code for these graphs.

	Line	Square	Star	Flag
R^{tot}	20.000000000000008	10	18	12.666666666666658

Figure 2.8: Outcome Effective Resistance for denoted graphs Figure 2.9

Adjusting our code for the undirected graphs to include weighted graphs as well, we found that the measure gives almost the same outcome (only a little difference one can ascribe to code being only an approximation) for corresponding undireced graph of each directed graph in Figure 2.9. Since the corresponding undirected graphs of the first two graphs in the figure are P4 and C4 with their weights divided into half, we notice that the outcomes of the measure are double the outcomes of P4 and C4. This might not be a coincidence.

We tried creating restrictions that would make the directed form of the measure either monotonically increasing of monotonically decreasing, but we came to a dead end. However, we still want to provide our ideas and their contradiction in order to encourage further tries. First, we want to note that whichever additional edges that also add an edge in the undirected graph with the same Total Effective Resistance, will yield a decrease of the measure. This is due to the proven monotonical decrease of the measure for that corresponding undirected graph when edges are added.

Secondly, we noticed that only the third example has additional edges to its undirected counterpart, making the undirected graph a complete one. Since that happened, we can not conclude that adding a directed edge will yield the addition of an edge to the accompanying undirected graph. Therefore we can not immediately conclude our research with a restriction that yields monotonical decrease.

Next, we tried finding out when extra edges are added in the undirected graphs that accompany the directed graphs in their outcome of the measure. Our idea was that it maybe was due to the fact that removing a node that was at the end of each chain of arrows in the graph, made the third graph unconnected. All of the other examples in Figure 2.9 show graphs in which extracting the/a node, a node one can get to from every other node, still leaves a connected graph. Sadly, adding a fifth node to the third example, accompanied by an edge from the fourth to the fith node, also yielded extra edges in the undirected counterpart. Removing that fifth node does not leave the graph unconnected and therefore concludes our research.

Finding a reason why extra edges are added to the undirected graphs with the same Total Effective Resistance as the directed ones, would lead us to the needed restrictions to get monotonical decrease of the measure for added edges that meet the restrictions. We encourage others to try and come up with the restrictions we search, if they exist.

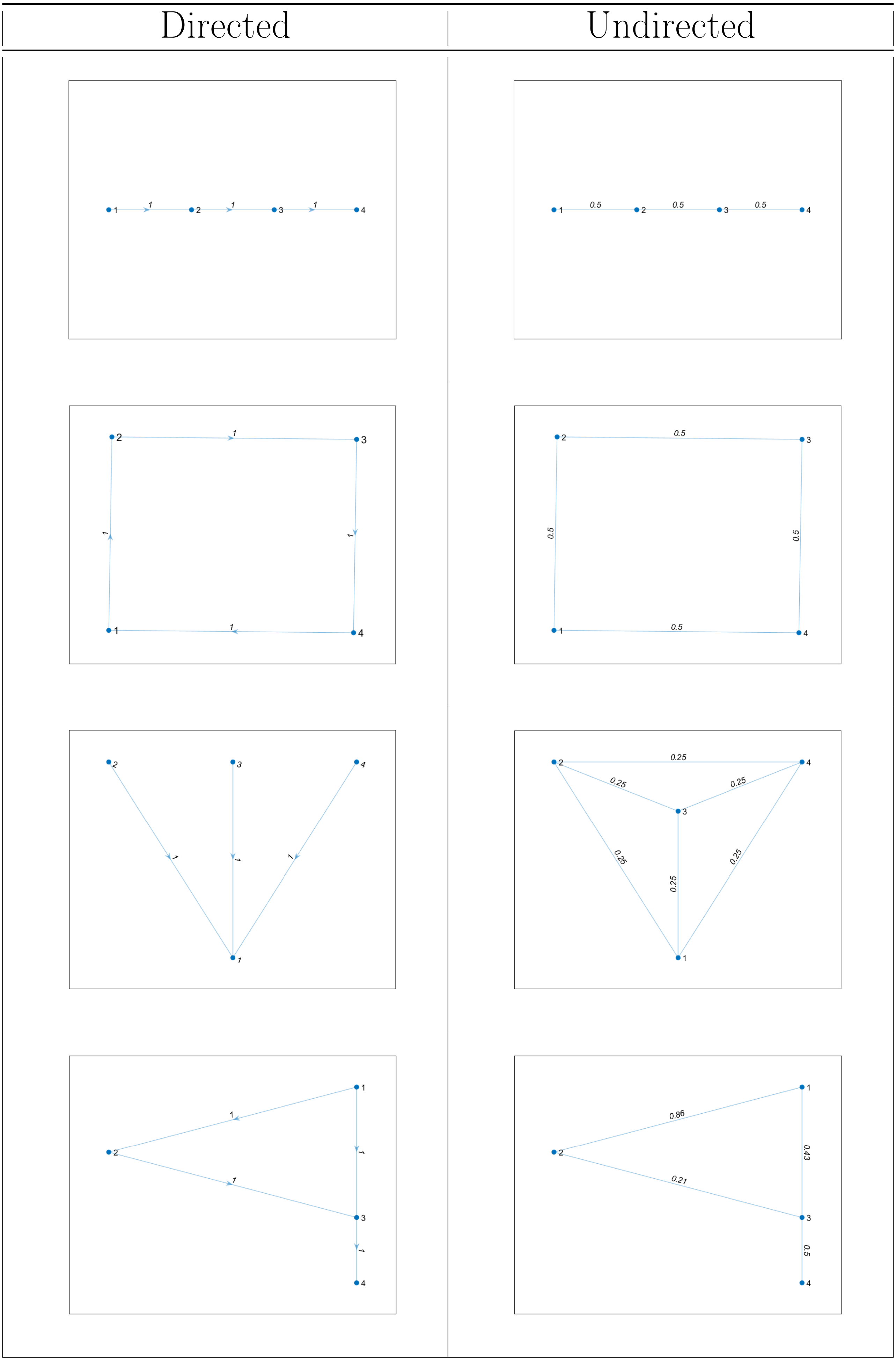


Figure 2.9: Graph representations of directed graphs and the undirected graphs with the same total Effective Resistance

Appendix A

Code used in this thesis

In this thesis, a lot of calculations had to be done. Since we did not come across immediately usable code, we decided on providing it ourselves. In this thesis there are references to this code, when we specified to have used it. If this code is what one seeks for further research, feel free to use this appendix as a guide to do so, since we will explain our code and how to use it, in this part of the thesis.

A.1 DYNER Measure

In the same order as being treated in the rest of this thesis, the DYNER Measure will get its turn first. The code has been built using Singer's[4] algorithm, that he displayed in his article. We followed Y.Singer's algorithm step by step and made it suitable for Matlab. There are no shorts takes, thus only basic code was used.

The code showed below in Figure A.1 is the general function used to calculate the DYNER Measure with Matlab. Ofcourse, there are three parameters that need explanation.

- When there are n nodes, the *set* parameter is the collection of the numbers 1 till n (in Matlab, that will be [1,2,...,n]).
- The parameters a and b are quite simple, for each edge one end is added to a and the other to b, also yielding a collection. For example, a graph with edges $\{1,2\}$, $\{3,4\}$ and $\{5,6\}$ will have the following input: $a = [1\ 3\ 5]$, $b = [2\ 4\ 6]$.

The reason why we chose to do this, is that it is fairly easy to alter this code to be able to calculate the DYNER Measure for a directed graph.

```matlab
function DYN = DYNER(a,b,set)
    G = graph(a,b);
    plot(G)
    d = distances(G);
    totDist = 0;
    totDistVia = 0;
    DYNPre = 0;
    for via = set
        set2 = set;
        set2(set2 == via) = [];
        for node1 = set2
            set3 = set2;
            set3(set3 == node1) = [];
            for node2 = set3
                totDist = totDist + d(node1 , node2);
                dsub = distances(subgraph(G,set2));
                if(node1 > via)
                    node1 = node1 - 1;
                end
                if(node2 > via)
                    node2 = node2 - 1;
                end
                totDistVia = totDistVia + dsub(node1, node2);
                if(node1 >= via)
                    node1 = node1 + 1;
                end
                if(node2 >= via)
                    node2 = node2 + 1;
                end
            end
            DYNPre = DYNPre + 1./totDist - 1./totDistVia;
            totDist = 0;
            totDistVia = 0;
        end
    end
    DYN = 1./DYNPre;
end
```

Figure A.1: Matlab code for DYNER Measure

A.2 The Total Effective Resistance

For the calculation of the Total Effective Resistance, two pieces of code have been used. One for the Effective Resistance of undirected graphs, the other for the directed graphs. Since there is no correspondence whatsoever, other that the measure they serve, we separately cover the two pieces of code. We want to add as a sidenote, that both pieces of code are easily expandible to covering weighted graphs as well.

A.2.1 Undirected ER code

The code in Figure A.2 originates from the way to calculate the Total Effective Resistance portrayed in Theorem 4, thus using the non-zero eigenvalues of the Laplacian. Since a computer only approximates, not fully calculates, the actual outcome, the code does not precisely yield a zero-eigenvalue. We decided on setting the smallest eigenvalue we found for each graph, equal to zero. The commentary in the code shows where we did this.

Again it is important to know what parameters are needed to plug into this function to get the code to calculate what one would want. a and b here are the same as in the DYNER Measure code. The parameter *number* just equals the number of nodes in the graph.

```
function ER = EFFRES(a,b, number)
    G = graph(a,b);
    plot(G);
    L = laplacian(G);
    full(L)
    eigen = transpose(eig(L));
    R = 0;
    smallest = 1;
    for n = eigen
        if abs(n) < eigen(smallest)
            smallest = find(eigen == n);
        end
    end
    eigen(smallest)=0; %zero EV not fully zero correction
    for n = eigen
        if n ~= 0
            R = R + 1/n;
        end
    end
    ER = number * R;
end
```

Figure A.2: Matlab code for Effective Resistance

A.2.2 Directed ER code

Last, but not least, the star of the show that should not be missing, is the code to calculate the Total Effective Resistance for directed graphs. This code far more difficult to comprehend than the other two, but very useful for calculating the Total Effective Resistance in directed graphs.

With commentary we tried to make the code as accessible as possible. The code followed the roadmap K.Fitch provided on the seventh page her article[2], which basically equals Definition 12 in this thesis. However, instead of the pseudoinverse we calculate the Total Effective Resistance. We did include the code to get the pseudoinverse and included it at the right spot as a commentary.

Again a and b are the same as in the DYNER code, also n equals *number* in the last piece of code (Figure A.2). The only 'new' parameter is textitone, which is an all-ones vector, with n entries.

```
function ERdirected = EFFRESDI(a,b,one, n)
%Calculate directed laplacian
    G = graph(a,b);
    h= plot(G);
    labeledge(h,1:numedges(G))
    A = adjacency(G);
    B = A;
    teller = 0;
    for i=1:size(B)
        for j=1:size(B)
            if B(i,j)== 1
                B(i,j) = -1;
                teller = teller + 1;
            end
        end
        B(i,i) = teller;
        teller = 0;
    end
    full(B);
%--------------------------------------
%Calculate Orthonormal basis for 1_n^perp
    ortho = null(one(:).');
    P = orth(ortho); %make sure it is orthonormal
    Q = P'; %make sure basis is in rows
%---------------------------------------
%Calculate reduced Laplacian
    Lnew = Q*B*Q'; %reduced laplacian
%---------------------------------------
%Solve Lyapunov Equation
    I = eye(n-1);
    minI= -1*I; %in order to use matlab function
    X = lyap(Lnew,minI);
%---------------------------------------
%Project to R^n*n
    Y = 2*Q'*X*Q;
%---------------------------------------
%Calculate pseudoinverse
    %undirL = pinv(Y);
%---------------------------------------
%Calculate EFFRES
    EfRes = 0;
    for i = 1:n
        for j = 1:i
            EfRes = EfRes + Y(i,i) + Y (j,j) - 2*Y(i,j);
        end
    end
    ERdirected = EfRes;
end
```

Figure A.3: Matlab code for directed Effective Resistance

Bibliography

[1] W. Ellens, *Effective resistance and other graph measures for network robustness,* 2011 Linear Algebra and its Applications, ISSN: 0024-3795, Vol: 435, Issue: 10, Page: 2491-2506.

[2] K. Fitch, *Effective resistance preserving directed graph symmetrization,* 2019 SIAM J. Matrix Anal.Appl., 40, pp. 49–65.

[3] D.J. Klein and M. Randic. *Resistance distance. Journal of Mathematical Chemistry,* 1993 12:81–95.

[4] Y. Singer, *Dynamic Measure of Network Robustness,* 2006 IEEE 24th Convention of Electrical Electronics Engineers in Israel, Eilat, 2006, pp. 366-370.

[5] V. Vos, *Methods for determining the effective resistance,* 2016 Master's thesis, Leiden University.

[6] . G. Young et al, *A New Notion of Effective Resistance for Directed Graphs—Part I: Definition and Properties,* 2016 IEEE Transactions on Automatic Control, Vol. 61, No. 7, pp. 1727-1736.

[7] . G. Young et al, *A New Notion of Effective Resistance for Directed Graphs—Part II: Computing Resistances,* 2016 IEEE Transactions on Automatic Control, Vol. 61, No. 7, pp. 1737-1751.

[8] F. Spieksma, *Storingsgevoeligheid van Netwerken*, 2013 Syllabus Vakantiecursis, pag.41–60, editor J.J.O.O. Wiegerinck, Publisher Platform Wiskunde Nederland, year 2016.