# ROBUSTNESS OF REAL NETWORK CONTROLLABILITY TO DEGREE BASED ATTACKS

by

David M. J. Lanigan


APPROVED BY SUPERVISORY COMMITTEE:

Dr. Justin Ruths, Chair

Dr. Alvaro Cárdenas

Dr. Tyler Summers

To Mrs. Kurtz and Mrs. Kane.

# ROBUSTNESS OF REAL NETWORK CONTROLLABILITY TO DEGREE BASED ATTACKS

by

DAVID M. J. LANIGAN, BS

THESIS

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN

MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

May 2018

# ACKNOWLEDGMENTS

I would like to thank my supervising professor, Dr. Justin Ruths, for his guidance and advice throughout this project, as well as for giving me the opportunity to study and learn about graph theory and network control theory and its applications. Additionally, I would like to thank Dr. Tyler Summers and Dr. Alvaro Cardenas for their assistance in completing this project, and for their valuable input into its content.

May 2018

ROBUSTNESS OF REAL NETWORK CONTROLLABILITY TO

DEGREE BASED ATTACKS

David M. J. Lanigan, MS
The University of Texas at Dallas, 2018

Supervising Professor: Dr. Justin Ruths

Real world complex networks vary greatly topologically from each other as well as from generated synthetic random networks. For example, social and biological networks typically have a community structure, while random networks do not. In order to investigate the robustness of the controllability of real networks to attacks on its edges, five different attacks based on the degree of the nodes were levied at common real-world networks systematically by removing either 2% or 5% of the edges, in steps, until 90% were removed. It was then investigated how well these real networks retain their controllability, especially in comparison to Erdos-Renyi and Barabasi-Albert synthetic networks. In particular, the question of how effective attacks focusing on destroying edges with a high source node in-degree and a high target node out-degree, performed in comparison to attacks focused on edges with high betweenness centrality, was reviewed. It was discovered, that in contrast to results with synthetic networks, for many real networks the betweenness attack performed worse than the in-out attack after a certain number of edges were removed. By observing how high density and community structure affect the ability to retain control over the network after these two attacks, an explanation for this

may be assembled. In addition, the difference between the potency of these two attacks, while network controls were fixed to nodes or allowed to move to more optimal input nodes, was studied.

TABLE OF CONTENTS

LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

The goal of this research is to study the effects that deliberate and random edge removal has on the controllability of real, complex networks. By applying different strategic percolation methods, based on edge properties, to various real-world complex networks, the effectiveness of each attack may be evaluated. In prior research work, Thomas et al. [1], showed that for several of the most popular synthetic networks, an attack or percolation directed at the edges with highest betweenness, caused the controllability of the network to deteriorate most quickly out of any attack attempted. An important question that was asked in this paper, was whether the effectiveness of the betweenness still held for real networks. In particular, the question of whether this attack's effectiveness would remain strongest even while operating upon networks with a community structure, was of interest.

It was discovered that the performance of many of the attacks differed greatly when applied to real networks rather than synthetic. This, in part, was expected. However, less expected and more interesting, was the large number of networks analyzed that were highly resistant to the high betweenness edge attack. In total 44% of the real networks that were analyzed were shown to be more resistant to the betweenness attack than to the in-out attack after 90% of edges were removed from the network, when the effectiveness of the attack was quantified by the parameter $\Psi$, a parameter invented for this quantification and defined within the paper.

The layout of the paper is as follows. Firstly, in chapter two basic graph theory and an introduction to complex networks is presented in order to give the reader and understanding of

the basics behind the methods used in the paper, especially as it pertains to the controllability of the networks and the attacks used upon them. Next, in chapter three control theory is discussed, explaining how structural controllability offers the tools to enable categorizing controllable and uncontrollable networks. Additionally, an explanation of robustness of network control is discussed as well as the measures used to quantify it. In chapter four, the methods and the results of the current study are presented and lastly in chapter five conclusions on the results are offered.

## CHAPTER 2

## GRAPH THEORY AND COMPLEX NETWORKS

In graph theory, a graph, also called a network, is a structure made up of vertices, often called nodes,[1] which are connected to one another by edges. Notationally, graphs are written as: $G = (V, E)$, where $V$ and $E$ are sets of vertices and edges, respectively. The symbols $N$ for the number of nodes and $L$ for the number of edges, is also used. A graph may be either undirected or directed[2]. If it is undirected there is no distinction between the connections linking any two nodes. If it is directed, then the edge between any two nodes has an orientation, either toward or away from the node. This is similar to a positive or negative vector. Following the analogy of the vector, the edges of a graph can be weighted with a value that describes a sort of "magnitude", which can be positive or negative. Unlike vectors, however, this magnitude does not necessarily affect the direction of the edge.

Graph theory is useful for linking together things of like nature, which have a rule between them that describes their interactions. Constructing and analyzing networks of these entities can help to explain these interactions. The graphs can describe many different things from how airports connect and road ways to social networks and neural nets. Complex networks or graphs with non-trivial topological features, prove to be very interesting to study, because the interactions they describe would be difficult to understand without the analytical methods afforded by graph theory.

---

[1] Throughout this paper the terms, graphs and networks, as well as, nodes and vertices, are used interchangeably.

[2] If it is directed, the graph is often referred to as a *digraph*. This terminology is used in this paper.

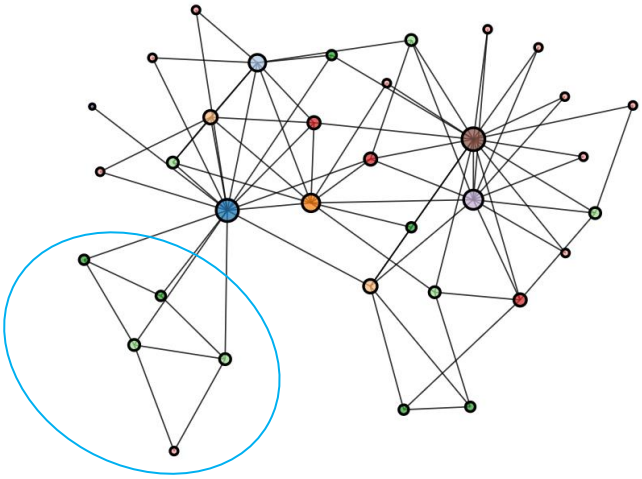


**Figure 1: Undirected Karate-Club Network.**

In addition, it is possible to create synthetic networks. Erdos-Renyi, Barabasi-Albert and the random partition graphs are examples of synthetic networks used in this paper and are described further in section 4.1.

## 2.1 Adjacency Matrix and Node Properties

Any graph can be represented by a useful abstraction known as the adjacency matrix. The adjacency matrix is a n by n matrix that we can call $\boldsymbol{A}$. Each element in $\boldsymbol{A}$ is defined as such for undirected networks: $\boldsymbol{A}_{ij} = 1$,[3] if there is an edge between i and j and $\boldsymbol{A}_{ij} = 0$ if there is no edge between $i$ and $j$. For directed networks there is a value only if there is an edge from $i$ to $j$, given $A_{ij} \neq 0$. Thus, the adjacency matrix is a simple and convenient way of describing the networks

[3] If the network is weighted, instead of the value 1, the weight of the connected between the two nodes is substituted.

"wiring diagram". Additionally, one can see that the adjacency matrix of an undirected network is always symmetric, but that of an undirected network is usually not. Throughout this paper, we deal only with directed networks.

A node in a digraph is said to have degree $k = k_{total} = n$, where $n$ is the number of edges that are either pointing towards or away from it and also represents the number of "neighbors" or adjacent nodes it has.[4] The node also has a in and out degree, where the out-degree ($k_{out}$) is the number of edges that point away from the node (the number of edges of which the node is a *source*) and the in-degree ($k_{in}$) is the number of edges that point toward the node (the number of edges to which the node is a *target*). In a digraph, it is therefore possible for each edge to have a *source* and a *target* node. The degree distribution of a network holds the total, in or out degree information for each node in the network.

## 2.2 Communities

When nodes in a network can be separated into distinct groups, it can be said the network has a community structure and the grouped nodes are communities. In figure 1, an example of a community is circled in a light blue circle. It is common for real networks to have at least some form of community structure, especially social or biological networks [2], [3]. This is distinct from most random networks who usually do not [4]. Therefore, the effect that this structure has on the ability of a given real graph to resist certain attacks, was an interest.

[4] In this paper $k$ will be used to denote degree. Average degree $\langle k \rangle$ is also used as an important parameter.

Communities can be classified as being either *strong* or *weak*. This is done using two degree based parameters inherent to nodes inside a community. One describes the number of neighbors the node has internal to the community ( $k_{int}$) and one describing the number of neighbors external from the community ($k_{ext}$ ). These can also be represented as percentages: $p_{int,i} = \frac{k_{int,i}}{N_i}$ , where $N_i$ is the number of nodes in the $ith$ community and $p_{ext} = \frac{k_{ext}}{N}$. Graphs with strong communities are defined as having, on average, $k_{int} > k_{ext}$ and graphs with weak communities as having, on average, $k_{int} < k_{ext}$ [5].

### 2.2.1 Betweenness Centrality

The edges between communities can often be characterized by a property called high *betweenness centrality* [6]. Centrality measures are important in network analysis because they are used to gauge the relative importance of an edge in the overall network structure. The betweenness centrality of an edge can be defined as the fraction of shortest paths between any two nodes in a network that run through that edge. Thus, high betweenness edges are often thought of as "bridges" between nodes in a network, which lends to the idea that these edges may be important for network control [7]. The high betweenness edge characteristic becomes increasingly apparent if the network has a well-defined community structure. The reason for this is, that, as the network is separated into its communities, usually only a few edges are left between the communities as per the definition of communities. Therefore, these edges bridge the nodes between the communities and will naturally have high betweenness.

## 2.3 Other Network Properties Related to Control

Certain other network properties that are important to understanding control of complex networks, are now discussed. Below are some important definitions that relate to control that will be useful later.

*Definition 1*: A **path** in a network is any sequence of vertices such that every consecutive pair of vertices in the sequence is connected by an edge in the network. For directed networks, the path must follow the correct direction [6].

*Definition 2*: A **cycle** in a directed network is a closed loop of edges with the arrows on each of the edges pointing in the same way around the loop [6].

*Definition 3*: A graph G contains a **dilation** iff there is a subset of vertices, say $S$, with neighboring vertices, say $T(S)$ such that $|T(S)| < |S|$, where $|T(S)|$ and $|S|$ denote the cardinality of $T(S)$ and $S$, respectively [8].

*Definition 4*: A node in a digraph is **inaccessible** if there are no edges directed toward it. Inaccessible node cannot be reached by a control; therefore, a control must be attached directly to it [9].

*Definition 5*: A **subgraph** of a graph $G(V, E)$ is a graph $G(V_1, E_1)$ where $V_1$ is a subset of $V$ and $E_1$ is a subset of $E$ and **a spanning subgraph** is a subgraph with the exact same vertex set as the original graph [6].

# CHAPTER 3

# COMPLEX NETWORK CONTROLLABILITY

Complex networks of entities and systems are all around us. One can find examples of this in the internet, with webpage interconnectedness forming a network, or in social groups, where friendships or acquaintanceships form a network. Modeling the interconnectedness of these different entities or systems can shed important light on them. Some examples of networks, such as those representing cell metabolic systems or supply chains, may describe systems that it is desirable to control. For this reason, the question must be asked: how does one control complex networks? And more specifically, what are the minimum number of controls necessary to control them?

As was discussed earlier, networks consist of edges and nodes, where the edges describe a relationship between nodes. In networks modeled for control purposes, the state of the nodes is what is desired to be controlled and the edges describe the effect nodes have upon their neighbors. Modeling large networks with complex dynamics seems like a near impossible task. However, this task is made easier by estimating the dynamics as linear. A simplification of this magnitude is permissible since the goal here is only to define whether or not the network is controllable. Additionally, it presents a foundation to move to more complicated analysis.

## 3.1 Modeling Complex Networks with Linear Dynamics

By estimating our network as an LTI system, linear control theory can be applied. If this is the case, then the dynamics of the system can be described using the following well known equation [10].

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \qquad \textbf{Equation 1}$$

We can use this equation to detail the dynamics of a complex system by defining $\boldsymbol{x}(t) \in \mathbb{R}^n$ as the vector describing the state of each node in the network. $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix discussed earlier containing the node interaction information, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is a vector describing input signals to the system and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ matrix containing the controls needed to fully control the system. In the context of control a positive weight signifies a excitation and a negative weight signifies inhibition [9].

If we combine the nodes and the controls, we can create a different, control-augmented graph $G(\boldsymbol{A}, \boldsymbol{B})$, the combined graph of both matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, that encompasses all the information of the fully controllable system. Where the nodes are equal to $n_{total} = n_A \cup n_B$ and the edges are equal to $E_{total} = E_A \cup E_B$.

Using the LTI definition of the system described by the state space equation, it is possible to define whether or not the network is fully controllable by a defined set of directly controlled nodes, using Kalman's controllability rank criterion [10], which states the system is controllable iff:

$$Rank([\boldsymbol{B}, \boldsymbol{A}\boldsymbol{B}, \boldsymbol{A}^2\boldsymbol{B}, \dots, \boldsymbol{A}^{n-1}\boldsymbol{B}]) = n \qquad \textbf{Equation 2}$$

Although this is a convenient and elegant way to uncover the controllability of the network, in real-life scenarios it is difficult or impossible to use. This is for two primary reasons, firstly, the network may have edge weights that are indeterminable, difficult to know, or these weights may be constantly changing. Secondly, it is computationally challenging for large networks to compute. To circumvent these problems, the concept of structural controllability may be applied.

### 3.1.1 Structural Controllability

The outline for structural controllability as it applies to complex networks was defined in the 1971 work by Lin [11]. It lays out a route to circumnavigate the issue of defining the reachability of a network algebraically, and allows for controllability to be established for even highly complex networks. Lin's structural controllability theorem gives the necessary and sufficient conditions for structural controllability.

Two graphs can have equivalent structure if for every fixed zero in $G_1(\boldsymbol{A}, \boldsymbol{B})$, there is a fixed zero in $G_2(\boldsymbol{A}, \boldsymbol{B})$ and vice versa, even if the non-zero values in both graphs are only known approximately. Structural controllability comes from the idea that if there is a controllable system where all parameters are known, it can be assumed that a second system, that has parameters that are approximately equal to those of the first system, is also controllable.

Therefore, to implement the idea on a system with matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{x}, \boldsymbol{u}$, define a second adjacency matrix $\boldsymbol{A_\alpha}$ to be a structurally similar matrix to $\boldsymbol{A}$. Meaning, for every zero value in $\boldsymbol{A}$ there is a zero value in $\boldsymbol{A_\alpha}$, and for every non-zero value in $\boldsymbol{A}$, there is a value $\alpha$ in $\boldsymbol{A_\alpha}$. If $\boldsymbol{A}$ is a controllable system under the Kalman rank condition, then $\boldsymbol{A_\alpha}$ can be shown to be controllable for almost all $\alpha$ [11]. This solves the issue of needing to know the weights of the edges of the digraph exactly to define whether it is controllable or not. In addition, by proving that certain subgraphs are always structurally controllable, it can be shown that through the union of a set of these subgraphs, which span all the nodes in the network (a second *spanning* subgraph), the network can be controlled by a combination of the input vertices, which also happens to be the minimal set of controlled nodes. Therefore, the problem then becomes finding these set of

subgraphs and the minimum set of controls that fully control this union of subgraphs. This problem is solved through maximum matching.

### 3.1.2 Maximum Matching

As described above, when attempting to discover whether a complex network is fully controllable or not, it is necessary to find the set of directly controlled nodes that fully control the network. Because linking one control to each node would be trivial, the problem becomes finding the minimum set of controls. These are found through the process called maximum matching. Maximum matching finds the largest set of nodes that can be uniquely paired amongst themselves using edges in the network. A commonly used algorithm for this is the Hopcroft-Karp algorithm for bipartite graphs [12]. This algorithm is frequently used because its worst-case scenario run time is low, only $O\left(L\sqrt{N}\right)$. Although it is designed for bipartite graphs, it can be used on directed graphs by converting them into bipartite graphs.

### 3.1.3 Hopcroft-Karp Algorithm

The Hopcroft-Karp algorithm finds the maximal cardinality[5] matching of an unweighted bipartite graph. This is accomplished by extending augmenting paths, between the matched and unmatched nodes, which are called free nodes, because they are not in the matching. An augmenting path starts at a free node and ends at a free node.

The steps of the Hopcroft-Karp algorithm are as follows:

1. Initialize $M$, the matching, as the zero set

[5] The cardinality of a set is a measure of the number of elements in the set.

2. Use breadth first search to build an alternating level graph, rooted at unmatched vertices in set 1 of the bipartite graph
3. Use depth first search to augment the current matching $M$ with the maximal set of vertex disjoint shortest-length paths
4. Repeat the above steps until there are no more augmenting paths.

For a directed graph, the output of the maximum matching is a set of edges.

### 3.2 Graph Control Structure: The Spanning Cacti

Once the matching is defined, it can be used to design the control structure of the graph. The minimum number of inputs necessary to render the graph controllable, is related to the maximum matching of the digraph. The *minimum inputs theorem* states that the number of directly controlled nodes needed to fully control a network is one, if there is a perfect matching for the graph, or it is equal to the number of unmatched nodes with respect to any maximum matching, if the matching is not perfect. The nodes to be directly controlled are therefore the unmatched nodes [9]. Thus, the maximum matching defines a set of *paths* and the *cycles* that form the basis of the spanning cacti for the network. Path and cycles with controls attached to them are the pre-defined structurally controllable subgraphs spoke of earlier, and the combination of them make up the spanning subgraph called a cacti. Therefore, using the following definitions from Liu et al. [9], we can define the controllability of a complex network.

*Definition 6*: A **stem** is a path originating from an input vertex (directly controlled node). The initial vertex of a stem is called the root of the stem.

*Definition 7*: A **bud** is an elementary cycle with an additional edge that ends, but does not begin, in a vertex of the cycle. This additional edge is called the *distinguished edge*.

*Definition 8*: A **cactus** is a subgraph defined recursively as follows. A stem is a cactus. Given a stem $S_0$ and buds $B_1, B_2, \ldots, B_l$, then $S_0 \cup B_1 \cup B_2 \cup \ldots \cup B_l$ is a cactus if for every $i$ in $(1 \leq$

$i \leq l$) the initial vertex of the distinguished edge of $B_i$ is not the top of $S_0$ and is the only vertex belonging at the same time to $B_i$ and $S_0 \cup B_1 \cup B_2 \cup ... \cup B_{i-1}$. A set of vertex-disjoint cacti is called a cactus.

Using the above definitions, Lin was able to define the controllability theorem below.

***Theorem 1*:**
The following three statements are equivalent:

1. A linear control system $G(\boldsymbol{A}, \boldsymbol{B})$ is structurally controllable.
2. 
    a. The digraph $G(\boldsymbol{A}, \boldsymbol{B})$ contains no inaccessible nodes.
    b. The digraph $G(\boldsymbol{A}, \boldsymbol{B})$ contains no dilations
3. $G(\boldsymbol{A}, \boldsymbol{B})$ is spanned by a cacti [9].

This theorem allows the controllability problem to be reduced to finding the cacti and therefore the minimal set of directly controlled nodes. Therefore, as described above, the problem is to perform a maximum matching on a graph, find the set of unmatched nodes, and assign controls to those nodes.

***Theorem 2*:**
The minimum number of inputs or the minimum number of controls ($N_C$) needed to fully control the network $G(\boldsymbol{A})$ is equal to the number of unmatched nodes with respect to any maximum matching. These unmatched nodes, become the input vertices [9].

If the maximum matching cardinality of a digraph is denoted by $|M|$, then the number of nodes attached directly to controls equals $N - |M|$, where $N$ is the number of nodes in $G(\boldsymbol{A})$. S was mention, if $|M| = N$, the matching is said to be *perfect* and any node can be chosen as a driver node [9].

### 3.2.1 Relation of $N_D$ to $\langle k \rangle$

As was shown by Liu et al. [9], the average degree of the network effects the minimum amount of controls needed to control the network. As can be seen in the equations below, for both BA and ER graphs, the percent controls ($n_C = \frac{N_C}{N}$), depends on the average degree ( $\langle k \rangle$ ) of the network, while for scale-free networks, the in and out degree ($\gamma_{in} = \gamma_{out} = \gamma$) is also a parameter [9].

$$n_{D,ER} \sim e^{-\langle k \rangle / 2} \qquad \textbf{Equation 3}$$

$$n_{D,BA} \sim \left[ -\frac{1}{2} \left( 1 - \frac{1}{\gamma - 1} \right) \langle k \rangle \right] \qquad \textbf{Equation 4}$$

Many of the real networks in this paper showed very high average degrees and following this general trend they are controlled by either one or a few controls. This resulted in interesting attack profiles.

## 3.3 Robustness of Control

The foundation for this work is two papers that study the effect of edge attacks on complex networks [13], [1]. The papers look into how node and edge based attacks effected the controllability of a variety of synthetic networks by evaluating them according to two different metrics of robustness. Although node based attacks are relevant, this paper, like Thomas et al. [1], focused on edge based attacks due to the possibility of more intriguing results.

### 3.3.1 Control Robustness Metrics

There are two ways used to describe the robustness of a networks controllability referred to by Thomas et al. [1]. These are the *control-based robustness measure* and the *reachability-based robustness measure*. The first is a measure that counts the number of new controls that must be added to a network to retain full controllability after a node or edge attack has occurred [14], [15]. The second measure was defined by Parehk et al. [13] and measures the number of nodes still controllable after an attack. This measure can be separated into two categories: for free controls and fixed controls. In the fixed controls case, the network has a defined set of controls on a defined set of nodes which do not change after an attack. In the free controls case, however, there is a defined set number of controls, but they are free to move to different input nodes, if it increases their reachability.

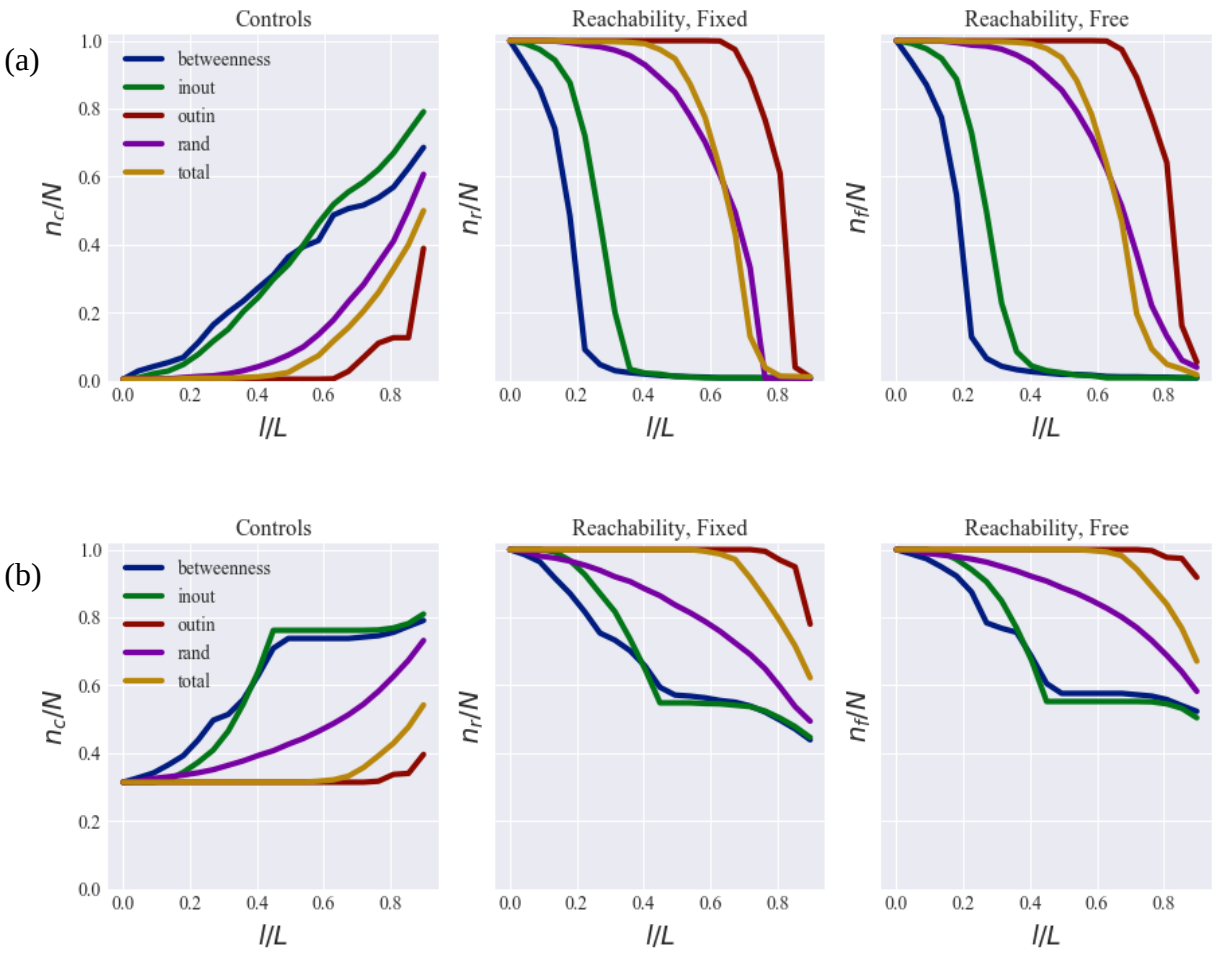



**Figure 2: Attack results for BA and ER networks.** The figure above shows how synthetic ER (a) and BA (b) networks with $\langle k \rangle$ = 6 respond to various attacks with three different control configurations, control count, free controls and fixed controls. The ordinate is the fraction of controls per total nodes ($n_C/N$) for the controls case and the fraction of reachable nodes for the fixed ($n_r/N$) and free ($n_f/N$) case, where $N$ is the number of nodes, $n_{c,r,f}$ is the number of controls or reachable nodes, $l$ is the number of edges removed and $L$ is the total number of edges. The abscissa is the fraction of edges removed. As can be seen, the betweenness attack is the most effective attack for both networks.

### 3.3.2 Control and Reachability based Robustness of Synthetic Networks

Control and Reachability based robustness measures were applied to Barabasi-Albert (BA) and Erdos-Renyi (ER) synthetic networks by Thomas et al. with interesting results. These measures were used to identify the effectiveness of various attacks on these two types of networks. Thomas et al. found that the in-out attack was the most potent of all degree attacks and made the case that the in-out degree measure, defined in section 4.1.1, was a proxy for the edge betweenness centrality measure. Indeed, as can be seen in figure 2, when the results of calculating both network's robustness measures, after removing 5% of the edges during each attack until 90% of the edges were removed, are plotted against the number of edges removed, the betweenness based edge attack degrades the network in a similar way to the in-out attack and was found to be more potent in almost all cases. This result showed that the in-out attack and the betweenness attack have strong similarities in the way they connect the network. However, it was not certain whether this inherent likeness would still produce similar control robustness results for all networks, especially those with a community structure.

To find the reachability of the set of controls for both the fixed and free case after each attack, requires a maximum matching procedure to be performed on a weighted bipartite graph formed from graph $G(\boldsymbol{A}, \boldsymbol{B})$, by the following methods found in [1] for the fixed controls case:

1. Remove nodes that cannot be reached by any control

For all $i, j = 1, \dots, N$ and $k = 1, .., m$:

2. Split the remaining nodes into a pair of positive and negative nodes: $x_i \rightarrow x_i^+, x_i^-$.
3. Add unit-weight edges $(x_i^+, x_j^-)$ if $(x_i, x_j) \in E$
4. Add unit-weight edges $(u_k^+, x_j^-)$ if $(u_k, x_j) \in E$
5. Add zero-weight edges for $(x_i^+, x_i^-)$ and $(u_k, x_j)$ (self-loops)
6. Add zero-weight edges $(x_i^+, u_k^-)$
7. Add a weight $W \geq |E|$ to all edges

This procedure is modified slightly for the free case by creating a new graph $G_2(\boldsymbol{A}, \boldsymbol{B})$ where each of the fixed number of original controls is linked to every node in the network by an edge. Because only one edge per control can be matched and all options for paths and cycles are available, the optimal configuration is produced.

# CHAPTER 4

# METHODS AND RESULTS

## 4.1 Methods

The methods used in this paper to investigate the networks robustness to controllability degradation are similar to those used in the paper mentioned in the previous section [1]. These are, to attack the network with different degree based attacks and analyze the networks robustness to these attacks using the three different robustness measures discussed in Chapter 3. The procedure is as follows: using one of the degree based attacks described in the next sub-section, a percentage of the edges of the network are selected for removal. For large networks 5% of the edges were removed each iteration and for smaller networks 2% were removed. This procedure was performed for each attack described in section 4.1.1, on 45 real world networks of varying "types", for example: social, neural or biological. In addition to attacking real networks, attacks were also performed on synthetic networks with communities. This was to analyze the effect communities had on the potency of the attacks, especially the edge betweenness attack.

**Synthetic Graphs Used:**

*Erdos-Renyi (ER):* A model for graphs that are synthetic random graphs with binomial degree distributions.

*Barabasi-Albert (BA):* A model for synthetic random graphs that generates a scale-free graph, which as a degree distribution that follows the power law.

*Random Partition (RP):* This graph is a random graph with defined communities. It is essentially an ER graph with community structure.

### 4.1.1 Definitions of Attacks

As mention, during an attack on the network, either 2% or 5% of the edges are removed each attack iteration. The edges are selected for removal based on the criteria listed below.

- **In-out degree attack**: The in-degree of the source node and the out degree of the target node are summed together and the edges with the highest values are removed first.
- **Out-in degree attack**: The out degree of the source node and the in degree of the target node are summed together and the edges with the highest values are removed first.
- **Total degree attack**: The total degree of the source and target node of the edge are summed together and the edges with the highest values are removed first.
- **Random attack**: Random edges are chosen for removal – this is also called edge *failure*.
- **Betweenness attack**: The edges with the highest edge betweenness centrality are removed first.

Degree based attacks were chosen to use because they are a local property of a node. Therefore, the attacker does not need to know the entire layout of the network in order to use these attacks effectively. The betweenness attack, however, is not locally based and therefore not as simple to use.

### 4.1.2 Categories of Networks

In order to examine how different categories of real networks responded to attacks, various network types were used in the analysis. A full list of the networks, grouped by network type, can be found in table 1 of this paper. It was noted, as can be reviewed in appendix A, which includes a full list of the network robustness results, that there is seemingly only a correlation between network type and degradation of controllability, insofar as the network structure is similar. Indeed, at times networks from different categories, could have more similarity with each other in their robustness to the attacks, than with their peers in the same category.

### 4.1.3 Software

Most of the work done in the paper was performed using the Zero-Effort Network (ZEN) library maintained by Derek Ruths, in concert with the netcontrolz complex network control library maintained by Justin Ruths. Additionally, the networkx library was used for certain functions, such as calculating edge betweenness centrality and generating the random partition network. All work was done in the Python programming language.

Zen documentation: http://zen.networkdynamics.org/

Netcontrolz documentation: http://justinruths.com/netcontrolz/

Networkx documentation: https://networkx.github.io/documentation/stable/#

## 4.2 Results

Perhaps the most interesting difference between the results of the attacks performed on real networks and those performed on synthetic Barabasi-Albert and Erdos-Renyi networks, is that in many of the real network attack responses, the betweenness attack was less effective than that of the in-out attack. As the structure of real networks can be vastly different from those of synthetic, this was a prominent question from the outset of the analysis. Also interesting, as can be seen in appendix A, where a list of the individual network attack results is found, was that with a significant number of networks, attacks such as out-in, random, and total were not significantly effective at reducing control at all. In addition, it was also noticed, that unlike synthetic networks, when network controls were changed from the fixed controls case to the free controls case, the betweenness attack's relative potency decreased more on average than that of the other attacks. Table 1 lists the networks that were analyzed, categorized by type, along with

some of their basic parameters. $N$ is the number of nodes in the network, $L$ is the number of edges, $\langle k \rangle$ is the average degree, $\rho_G$ is the density of the graph, $N_C$ is the minimum number of controls for complete controllability, and $n_C$ is the percentage. As can be seen there are many highly dense networks with low numbers of controls.

**Table 1: Properties of real networks used.**

| *Type* | *Name* | $N$ | $L$ | $\langle k \rangle$ | $\rho_G$ | $N_C$ | $n_C$ |
|---|---|---|---|---|---|---|---|
| Neural | celegans | 297 | 2345 | 7.9 | 0.027 | 49 | 0.16 |
| | macaque71 | 71 | 746 | 10.5 | 0.15 | 1 | 0.01 |
| | mac | 62 | 3844 | 62.0 | 1.016 | 1 | 0.02 |
| | mac95 | 94 | 2390 | 25.4 | 0.273 | 9 | 0.10 |
| | cocomac | 193 | 12051 | 62.4 | 0.325 | 4 | 0.02 |
| Transcription | yeast | 688 | 1079 | 1.6 | 0.002 | 565 | 0.82 |
| | ecoli | 418 | 519 | 1.2 | 0.003 | 314 | 0.75 |
| Intra-Organizational | manuf-frequency-reverse | 77 | 2228 | 28.9 | 0.381 | 1 | 0.01 |
| | manuf-famliarity-reverse | 77 | 2326 | 30.2 | 0.397 | 1 | 0.01 |
| | cons-frequency-reverse | 46 | 879 | 19.1 | 0.425 | 2 | 0.04 |
| | cons-quality-reverse | 46 | 858 | 18.7 | 0.414 | 1 | 0.02 |
| | physician-friend-reverse | 228 | 506 | 2.2 | 0.01 | 52 | 0.23 |
| Online Communication | email-Eu-core | 1005 | 25571 | 25.4 | 0.025 | 139 | 0.14 |
| | dnc-emails | 1891 | 5598 | 3.0 | 0.002 | 1500 | 0.79 |
| | polblogs-reverse | 1224 | 19025 | 15.5 | 0.013 | 436 | 0.36 |
| | one_mode_messgage | 1899 | 20296 | 10.7 | 0.005 | 614 | 0.32 |
| | one_mode_char | 1899 | 20296 | 10.7 | 0.005 | 614 | 0.32 |
| Social Influence | physician-advice-reverse | 215 | 480 | 2.2 | 0.01 | 78 | 0.36 |
| | physician-discuss-reverse | 231 | 565 | 2.4 | 0.011 | 58 | 0.25 |
| | teacher-student | 60 | 94 | 1.6 | 0.027 | 35 | 0.58 |
| Social | prison | 134 | 4489 | 33.5 | 0.252 | 94 | 0.70 |
| | physician-friend-reverse | 228 | 506 | 2.2 | 0.01 | 52 | 0.23 |
| | freeman1 | 34 | 695 | 20.4 | 0.619 | 1 | 0.03 |
| | freeman2 | 34 | 830 | 24.4 | 0.74 | 1 | 0.03 |
| | freeman3 | 32 | 460 | 14.4 | 0.464 | 1 | 0.03 |
| | social1inter_st | 67 | 182 | 2.7 | 0.041 | 9 | 0.13 |
| | social3inter_st | 32 | 96 | 3.0 | 0.096 | 6 | 0.19 |
| Protein Structure | ps1 | 95 | 213 | 2.2 | 0.024 | 18 | 0.19 |
| | ps2 | 53 | 123 | 2.3 | 0.045 | 13 | 0.25 |
| | ps3 | 97 | 212 | 2.2 | 0.023 | 20 | 0.21 |
| Trophic | foodweb-baydry | 128 | 2137 | 16.7 | 0.131 | 29 | 0.23 |
| | foodweb-baywet | 128 | 2106 | 16.5 | 0.13 | 30 | 0.23 |
| | maayan-foodweb | 183 | 2494 | 13.6 | 0.075 | 99 | 0.54 |
| Animal Dominance | bison | 26 | 314 | 12.1 | 0.483 | 1 | 0.04 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | cattle | 28 | 217 | 7.8 | 0.287 | 3 | 0.11 |
| | sheep_dominance | 28 | 250 | 8.9 | 0.331 | 1 | 0.04 |
| | hens_dominance | 32 | 496 | 15.5 | 0.5 | 1 | 0.03 |
| | macaque_dominance | 62 | 1187 | 19.1 | 0.314 | 3 | 0.05 |
| Metabolic | figeys | 2239 | 6452 | 2.9 | 0.001 | 1906 | 0.85 |
| | protien_stelzl | 1706 | 6207 | 3.6 | 0.002 | 765 | 0.45 |
| Electrical | s208 | 122 | 189 | 1.5 | 0.013 | 29 | 0.24 |
| | s420 | 252 | 399 | 1.6 | 0.006 | 59 | 0.23 |
| | s838 | 512 | 819 | 1.6 | 0.003 | 119 | 0.23 |
| Infrastructure | opsahl-openflights | 2939 | 30501 | 10.4 | 0.004 | 872 | 0.30 |
| | openflights1 | 3425 | 37596 | 11.0 | 0.003 | 1149 | 0.34 |

### 4.2.1 Betweenness Centrality versus In-Out degree based attack

As was mentioned before, one result from the work of Thomas et al. was that the high edge betweenness centrality attack was more effective on ER and BA networks than the in-out degree attack. It was surmised that this is true due to the fact that the in-out attack is a proxy for the betweenness attack. This hypothesis was supported by the fact that the betweenness attack degraded controllability similarly to the in-out attack as well as outperformed it in most cases. It was of particular interest, therefore, to investigate whether this result held for real networks and was found not to be the case in general. To aid in reporting this finding, a relative effectiveness ( $\Psi_{1,2}$), is defined as follows:

$$\Psi_{\alpha_1-\alpha_2} = \frac{\left(\sum_{i=0}^{N}\left(\frac{n_{\alpha_1}}{N} - \frac{n_{\alpha_2}}{N}\right)\right)}{N} \qquad \textbf{Equation 5}$$

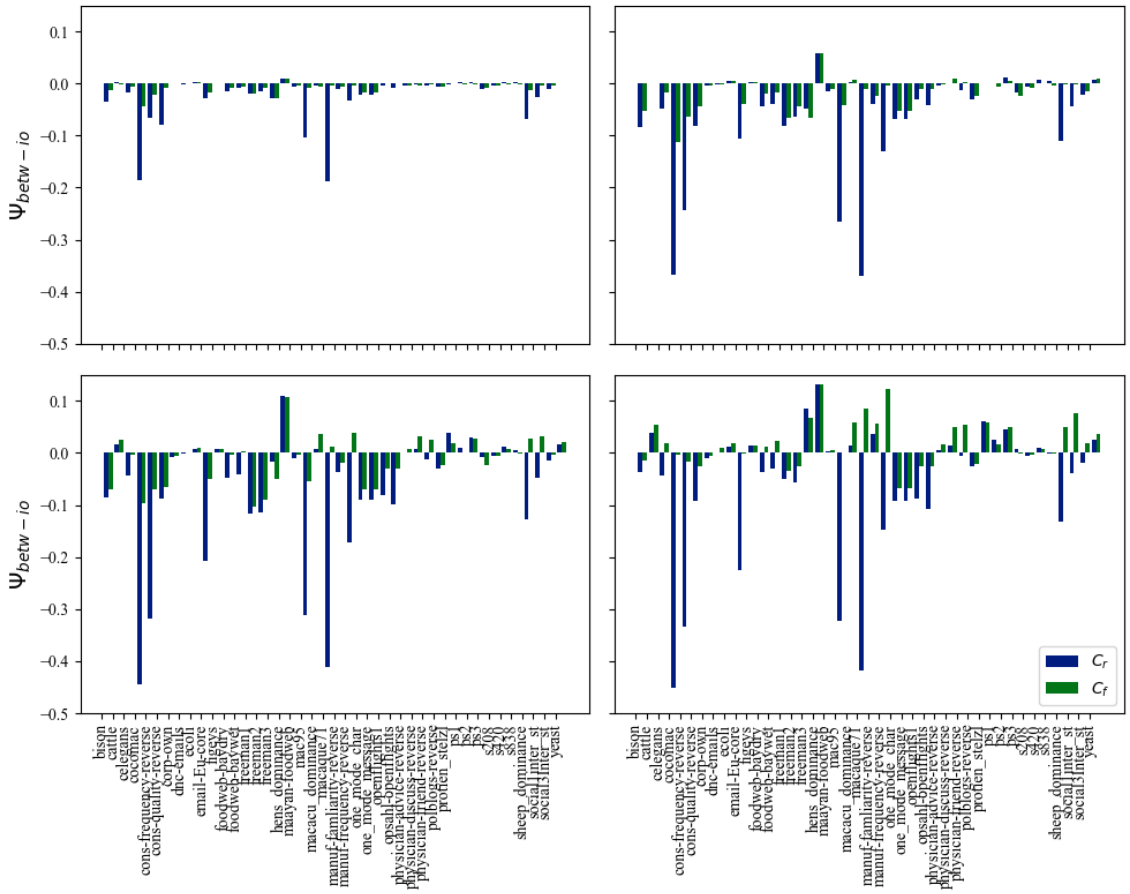



**Figure 3: Change in Ψ as attack progresses.** The figure above shows the value of Ψ at different edge removal steps during the attack process. From left to right, top to bottom the percentage of edges removed for the figures is 20%, 40%, 60% and 80%. The value of Ψ for both the fixed (blue) and free (green) cases are displayed for each graph. As the figure relates, as more edges are removed, in general, the betweenness attack becomes less effective in comparison to in-out attack.

Where $\alpha_1$and $\alpha_2$ are two different attacks and $N$ is the number of attacks on the network. The relative effectiveness is therefore a measure of which attack is reducing controllability more at each edge removal step. If the value of Ψ is less than zero, $\alpha_2$ is less effective at reducing control over all, and if Ψ is greater than zero, $\alpha_1$ is less effective at reducing control over all.

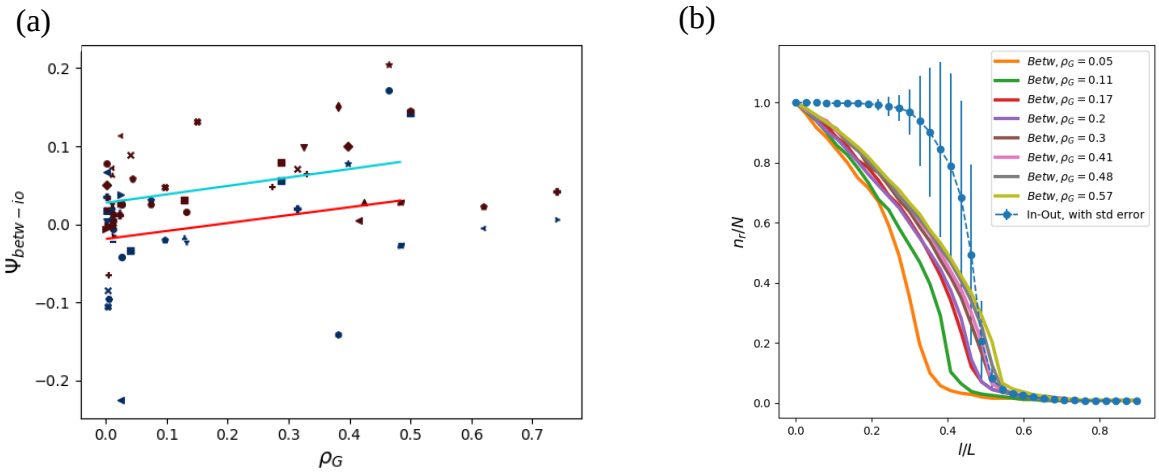



**Figure 4: $\Psi$ *vs* $\rho_G$ and effect of density on betweenness attack in ER networks.** The above figure (a), is a plot of $\Psi$ vs the graph density of each of the real graphs in both fixed (blue) and free (maroon) controls case for 90% edges removed. In the free case, graphs with a single controlled node where removed to prevent skewing of results. As can be seen, the least-squares fit trend line for both fixed (red) and free (light blue) increases for both sets at a rate of 0.15 $\Psi/\rho$ for free and 0.11 $\Psi/\rho$ for the fixed case. Figure (b) shows the betweenness attack plot of nine ER graphs with approximately 250 nodes each and densities varying from 0.05 to 0.57 plotted along with the in-out attack profile averaged over the nine graphs. This shows that as density increases, the betweenness attack curve converges to the in-out attack curve.

Throughout this paper, $\alpha_1$ is the betweenness attack and $\alpha_2$ is the in-out attack. As can be seen in figure 3, the betweenness attack is less effective than in-out attack for many of the real networks analyze after a certain amount of edges are removed.[6] In addition, the number of networks where betweenness is less effective grows as more and more edges are removed from the network. Also, when controls are made free rather than fixed, almost all networks retain controllability better while undergoing a betweenness attack at 90% edges removed. This was not the expected result following the work of Thomas et al. Therefore, it was of interest to examine why the

[6] This effect can also be seen, by viewing the individual network attack profiles in appendix A.

difference was so apparent in many of the networks. Further examination of network properties led to the conclusion that for at least some networks, *network density* and the existence of *community structure* may play a critical part.

**Effect of Network Density:**

Many of the real networks that displayed a reduced sensitivity to the high betweenness edge attack had notably higher density and a lower number of controls necessary for complete controllability (see table 1). It was surmised that this density which necessarily provides a more highly connected network, was behind the lessened effectiveness of the betweenness attack in relation to the in-out attack. As can be seen in figure 4 (a) there is a general trend which is characterized by a least-squares fit line, that shows that the high betweenness attack is becoming less effective as density increases for both free and fixed controls, and that this relationship increases when the controls are free. Indeed, this idea is supported by increasing the density of a synthetic ER graph and observing the change in the profiles of the betweeness attack and in-out attack curves. As can be seen in figure 4 (b), the betweenness attack curve tends to converge toward the in-out attack curve and at higher densities even becomes equal in its ability to degrade controllability of the network to the in-out attack. However, there is a large amount of variation in the results displayed in figure 4 (a) and, as can be seen in figure 4 (b), although increasing the density for ER graphs does *decrease* the effectiveness of the betweenness attack after a certain number of edges are removed from the network, it doesn't make the attack *less* effective than in-out overall. Therefore, there is another network characteristic that plays a part.

**Effect of Communities:**

The second network property that most likely plays a large part in reducing the effectiveness of the betweenness attack is the network's community structure. This is thought to be because the introduction of communities into the graph gives rise to a large amount of high betweenness edges that are not critical for control of the network (see figure 6 (a)). The effect that community structure has on a random network with a binomial degree distribution is shown in figure 5. In figure 5 (a), three different random partition graphs with varying numbers of communities are plotted along with a RP graph with a single community (essentially an ER graph) to use as a baseline. Each graph has 200 nodes and a different value of $p_{ext}$ or the percent of edges between communities. Each undergoes a betweenness and in-out attack and the $\Psi$ value is calculated and plotted against $p_{ext}$. As can be seen splitting the 200 node RP graph into a graph with two 100 node communities, has an immediate effect on the potency of betweenness attack. As can be seen the curve of the plot takes a peaked form, where, as the percent of edges between the communities is increased, the effectiveness of the betweenness attack is decreased, until a critical point is reach and the trend is reversed. This results in a "window" where the value of $p_{ext}$ is such that the ability of the betweenness attack to degrade network controllability is less than that of the in-out attack. As communities are added, this effect becomes more pronounced. Similarly, in figure 5 (b), the same effect can be seen, however, here three graphs with different $p_{int}$ values are plotted together. In this case, it can be seen that varying the $p_{int}$ value broadens the "window" of lowered betweeness effectiveness.

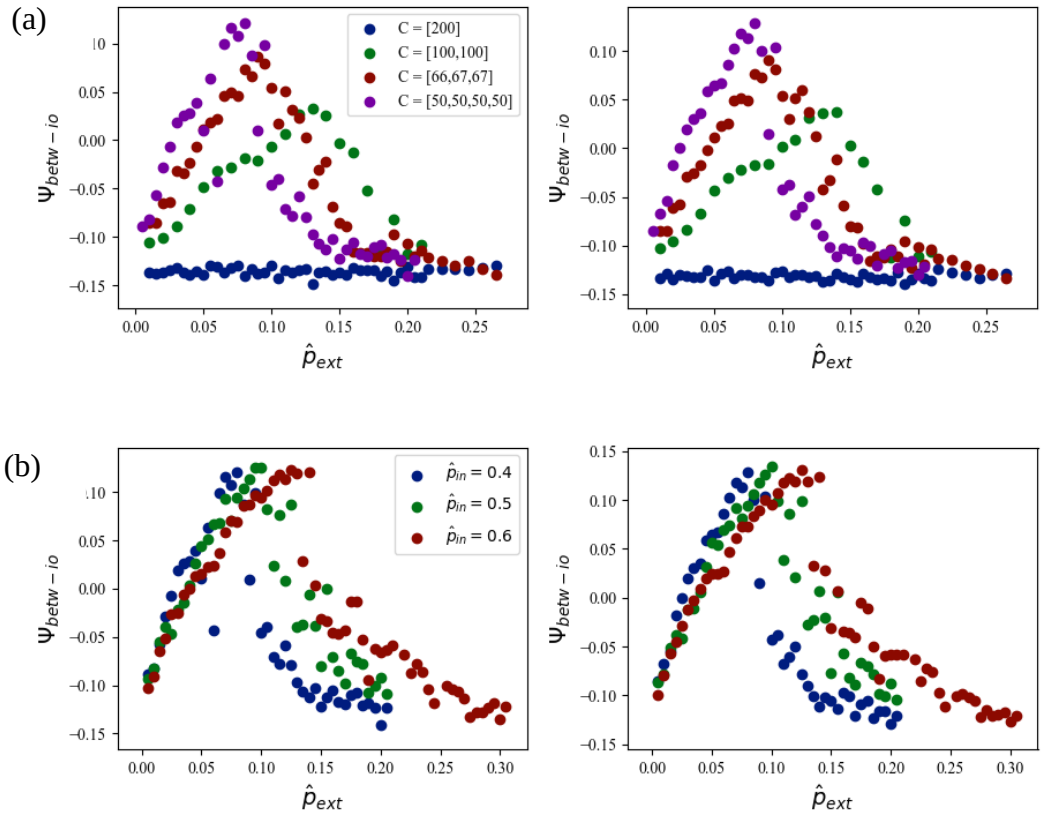



**Figure 5: $\Psi$ $vs$ $p_{ext}$ for different numbers of communities and different community densities.** The figure, where C represent a vector where the entries are the number of nodes in the communities and the length is the number of communities, shows the results of 3 sets of RP graphs with constant $p_{int}$ , as the percent of edges between the communities ($p_{ext}$) is varied. As one can see, an interval where the betweenness attack is rendered less effective than in-out attack exists. In figure (a) it can be seen that there is no such change in effectiveness for a graph with a single community, but if even two communities are defined in the graph, the interval appears. Additionally, as is seen in figure (b), more dense graphs have broader intervals then less dense graphs. For each figure (a) and (b), both fixed (left) and free (right) controls cases are plotted. As can be seen, there is little difference between them.

Additionally, although a reduction in the betweenness attack effectiveness can be produced by adding communities to low-density RP graphs, it does not cause betweenness attack to become less effective than in-out to any appreciable degree. For this reason, among other

possible factors, it was concluded that both density and community structure play an important role in the effectiveness of certain edge attacks.

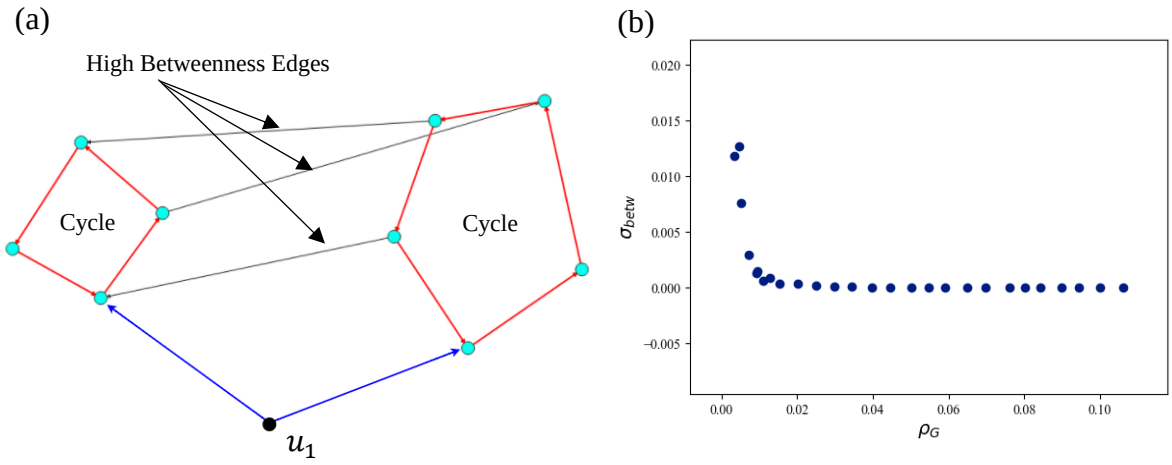



**Figure 6: Diagram of cycles controlled by a single control and a plot of standard deviation of betweenness centrality vs graph density.** The simplified graph in the figure (a) relates how a single control ( $u_1$) can control two communities via cycles while rendering the high betweenness centrality edges between them unimportant to control. Additionally, as the density of an ER graph of 400 nodes is increased (b), the normalized standard deviation of the betweenness centrality of the edges decreases and converges to zero.

The reasoning behind how communities lesson the strength of the betweenness attack can be visualized using figure 6 (a). As was mentioned high density requires higher average degree, which leads to lower numbers of controls necessary to control the network. Networks with very high density usually have a perfect matching and therefore have only a single control which attaches to cycles that form the control cacti. These cycles can be formed inside the communities and render the edges between the communities with high betweenness essentially useless for control purposes. Therefore, the removal of these high betweenness edges does not affect the controllability the way it normally would in community-less networks.

High density is thought to reduce the effectiveness of betweenness for a different reason. In graphs with high density, nodes are highly connected to one another and therefore on average

it takes only a few “steps” along any path to reach one node from another in the network. Therefore, inside the communities, the edges have similar betweenness and importance in node connectivity, which reduces the effectiveness of the betweenness attack. The “leveling out” of the average betweenness centrality value with high density graphs can be seen in figure 6 (b), where the density of an ER graph and the standard deviation of the normalized edge betweenness centrality are plotted together. This leads to an exponential convergence of the standard deviation to zero as density is increased.

The fact that the betweenness attack is rendered less effective for certain network is an interesting result, because it shows that for some networks, only local node knowledge is necessary for the most effective attack.

### 4.2.2 Free vs Fixed Controls

There was also seen to be a significant difference between the reduction in the effectiveness of the betweenness attack, in comparison to the in-out attack, when the controls were switch from free to fixed. For example, if controls are made fixed, roughly half or 44% of the networks analyzed were less sensitive to the betweenness attack than to in-out attack after 90% of the edges were removed. However, after controls were made free, this number rose to 86%.

When the network controls are allowed to be free rather than fixed, the effectiveness of every attack drops. However, the results from BA and ER analysis show that the percent reduction is essentially equal for all the attacks. However, for real networks, the betweenness attack is affected significantly more than the others as can be seen by figure 3.

For high density graphs, the explanation for this is simple, these graphs have very high degree and therefore a low number of controls which allows for the potential that these controls are isolated quickly by an attack. However, a low number of controls, even a single control, does not always mean controllability will be destroyed quickly, as is seen by the following two examples.

*Case 1 (macaque71):*

For the real network macaque71, $N_C = N - |M| = 1$. The cacti consists of a single stem and 19 buds. An examination of the plot of how the controllability degrades in appendix A, shows how the controllability of the network is completely destroyed within a few edge removal iterations. This is because the high betweenness attack quickly isolates the node the control is attached to from the rest of the network. However, when controls are free, this single control is allowed to move to different nodes and is therefore able to retain control of the nodes in the network fairly well.

*Case 2 (manuf-familiarity-reverse):*

For the real network manuf-familiarity-reverse, $N = |M| = 77$, therefore, a perfect matching is found and $N_C = 1$. However, in this situation, the single control links itself to each bud in the cacti via a distinguished edge, therefore there are 12 nodes driven by the one control. Despite having only a single control, the controllability of the network degrades in the fixed controls case in a manner similar to that of the free controls case. This is due to the fact that, after an attack, the new maximum matching of the network readily finds new cycles to control many of the nodes, and because the control is not permanently attached to a single node as is the case with a stem, it can re-attach itself to the new-found cycle and thereby retain control over much of

network very effectively. This gives the impression that the control "moves" as it constantly re-attaches distinguished edges to new cycles each attack iteration. Therefore, there is little difference between the degradation of control for fixed or free controls.

The apparent "movement" of a single control's distinguished edges to different cycles after an attack can shed light on a possible explanation for why *low* density networks with free controls have reduced sensitivity to the high betweenness centrality edge attack. Despite not having the assistance of high density to reduce the betweenness attack's effectiveness, allowing the controls of low density networks to freely move, gives these controls a chance to find new stems and cycles inside the community structure of the real networks. Therefore, as the betweenness attack progresses, focusing on edges between the communities, the cacti control structure is not affected as much by the attack and therefore the degradation of the networks controllability is slower.

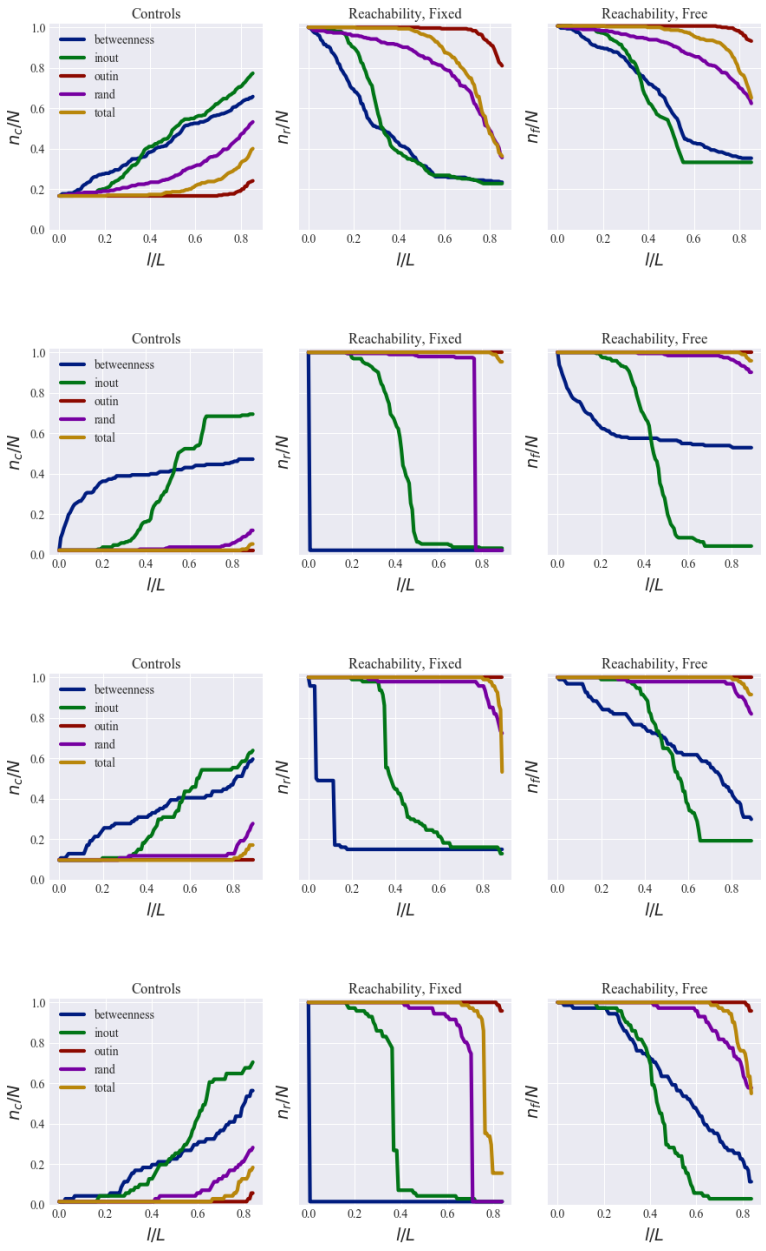
# CHAPTER 5

# CONCLUSION

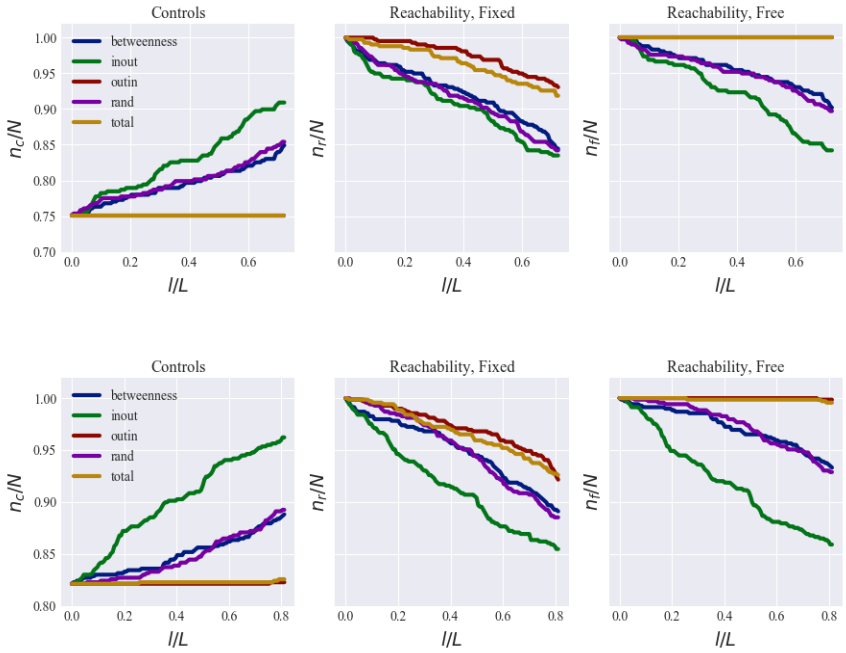
In conclusion, it was found for many real networks, the controllability of the network was effective very differently by certain attacks than for synthetic networks. This was most apparent in how the betweenness edge attack became less effective, after a certain number of edges were removed, than the in-out attack for many networks, showing that only local properties of the node are necessary for the most potent attack to be applied. This behavior was noted to have a possible relationship with both the density and the community structure of the network. In addition, for a substantial number of networks, attacks other than betweenness and in-out where mostly ineffective or had low effectiveness at reducing control, especially for those graphs with high density. Lastly, it was noted that the effectiveness of the betweenness attack suffered substantially more than other attacks when controls were set to be free rather than fixed. This behavior was most clearly showcased by networks with high density, who in many cases lost all controllability after only a few iterations of the betweenness edge attack, highlighting their vulnerability to these attacks.

# APPENDIX A

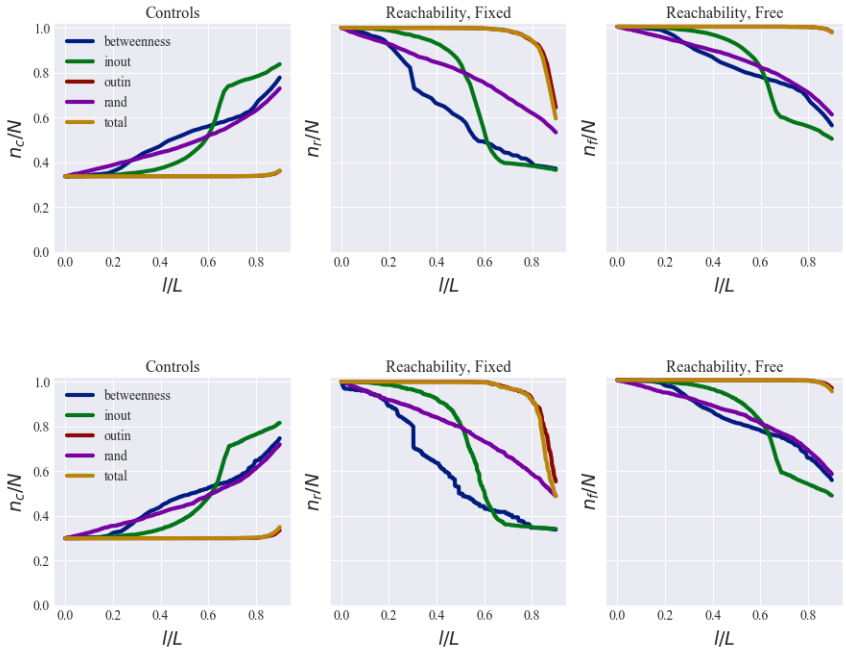
## NETWORK ATTACK RESULT FIGURES



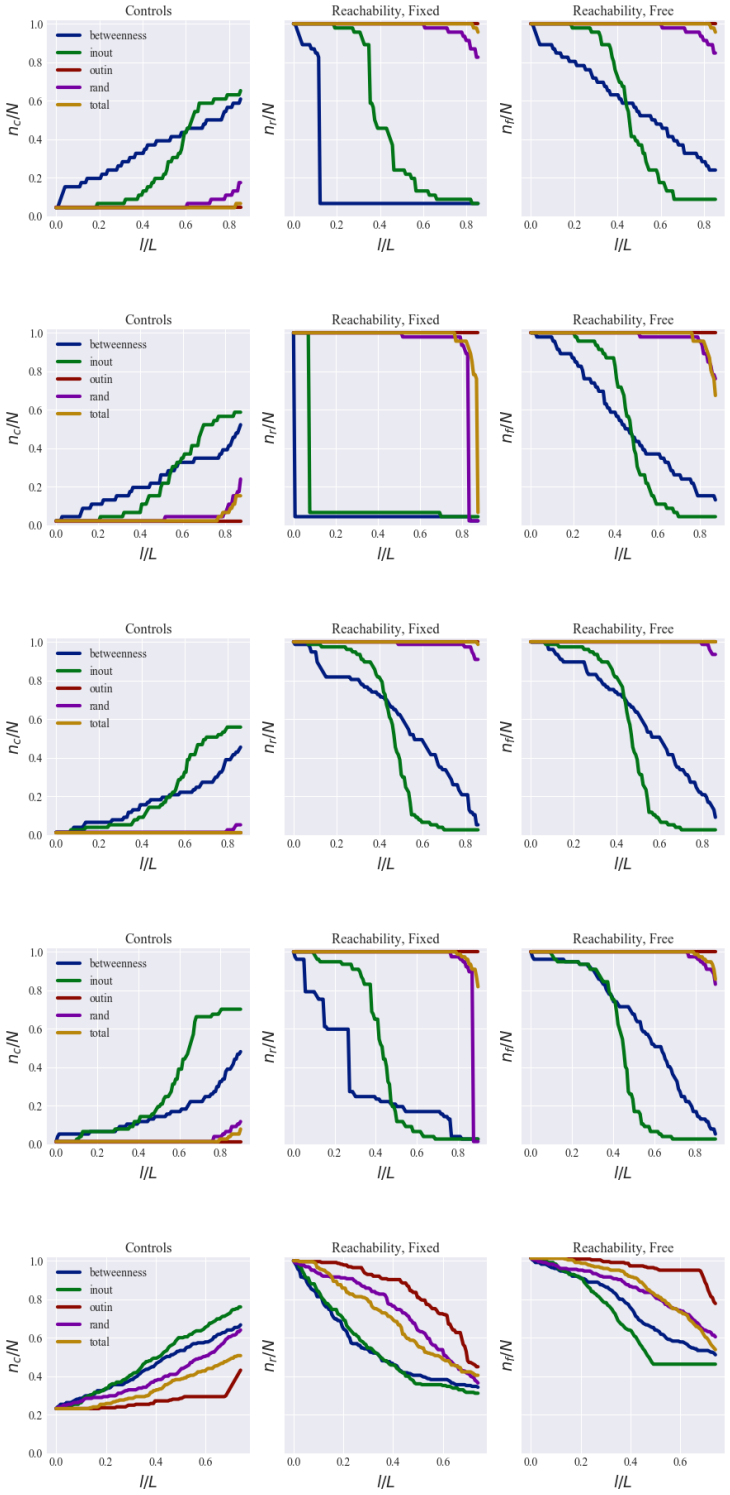

**Neural Graphs (top to bottom):** *celegans*, *cocomac*, *mac95*, *macaque71*.



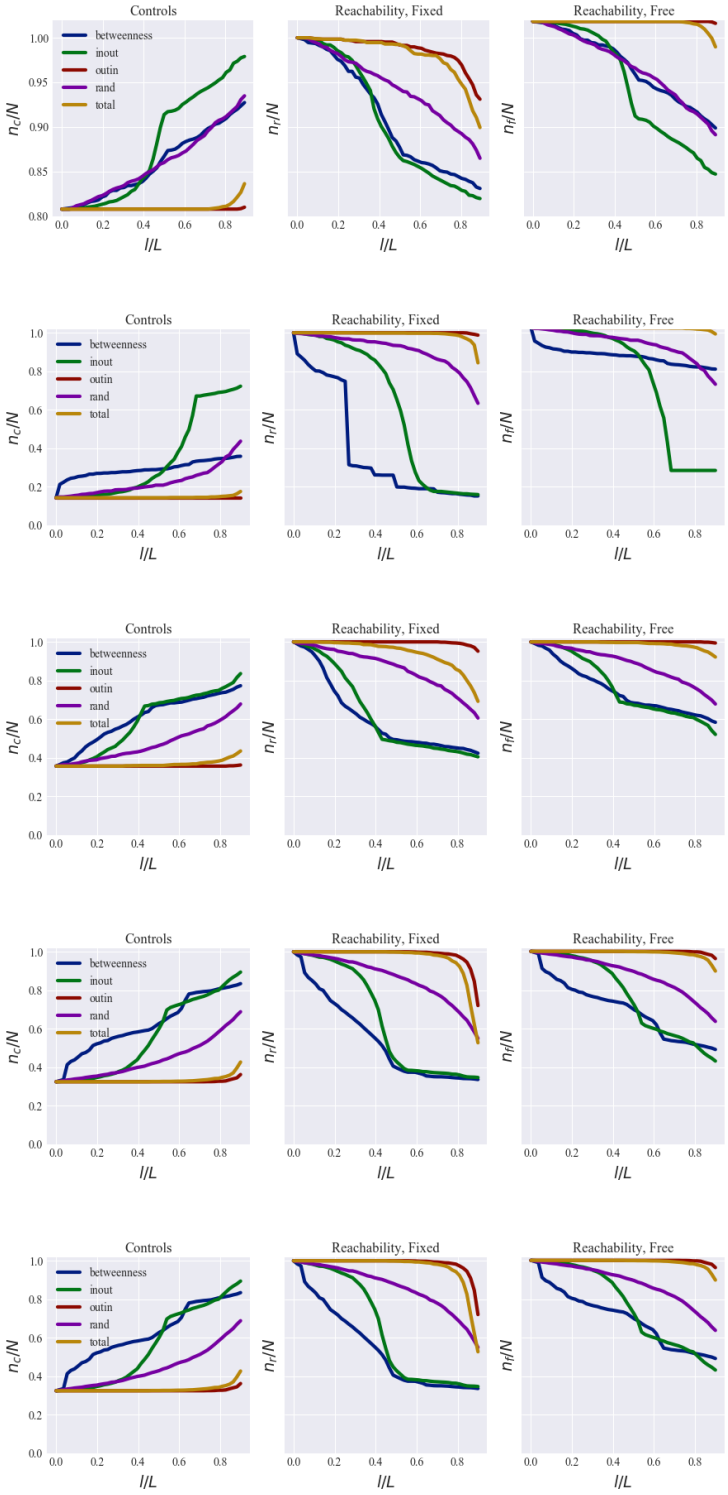

**Transcription Graphs (top to bottom):** *ecoli*, *yeast*.



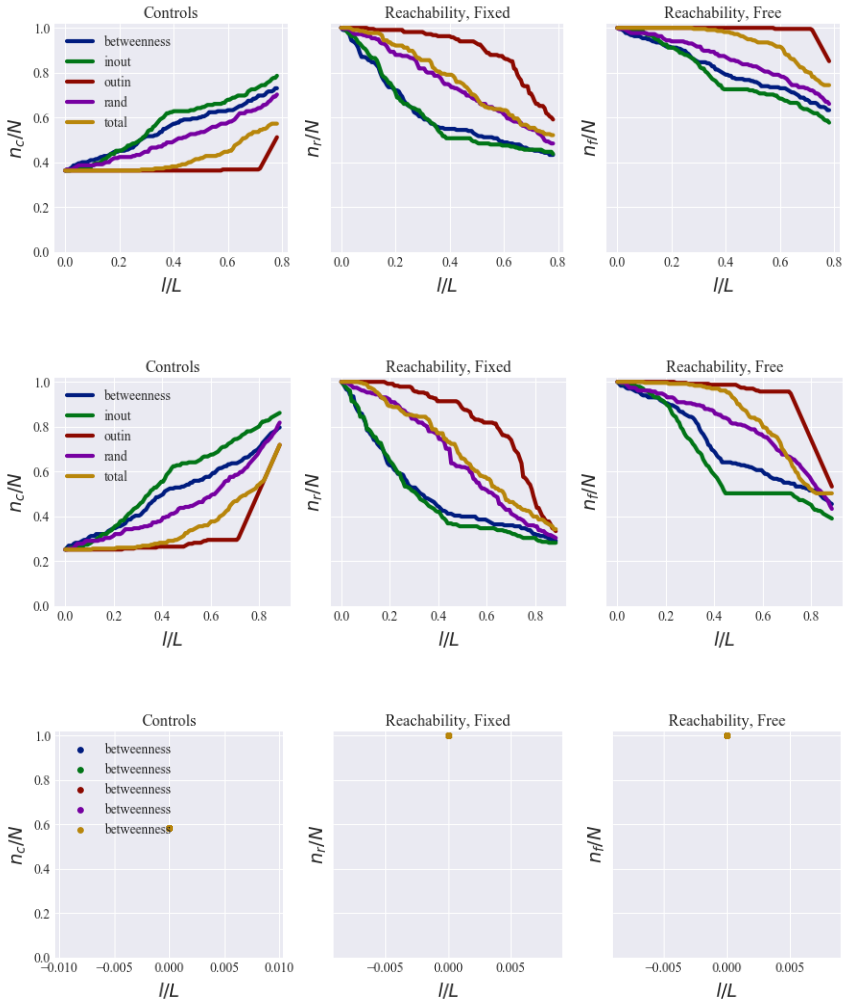

**Infrastructure Graphs (top to bottom):** *openflights1*, *opsahl-openflights*.



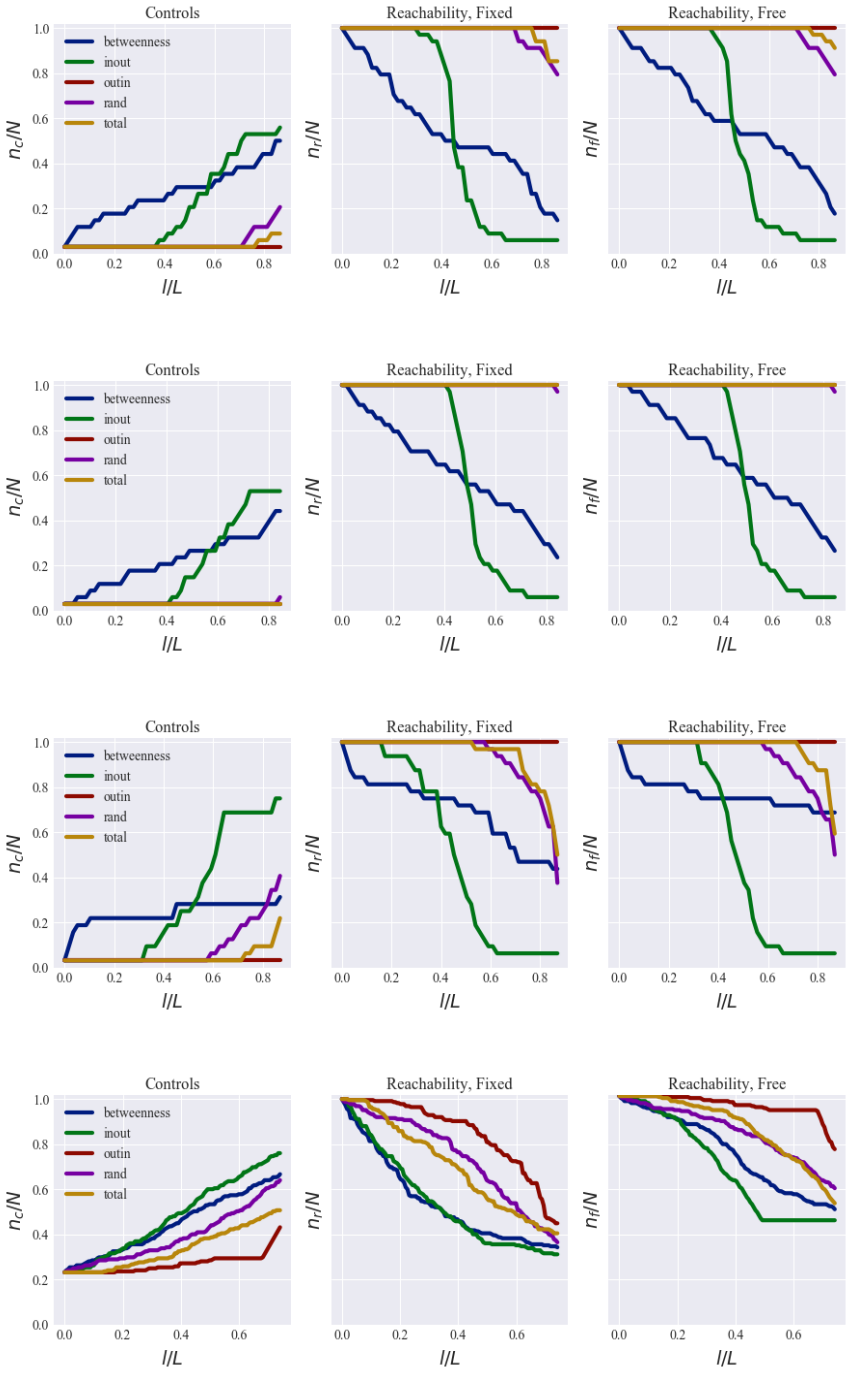

**Intra-Organizational Graphs (top to bottom)**: *cons-frequency-reverse*, *cons-quality-reverse*, *manuf-famliarity-reverse*, *manuf-frequency-reverse*, *physician-friend-reverse*.

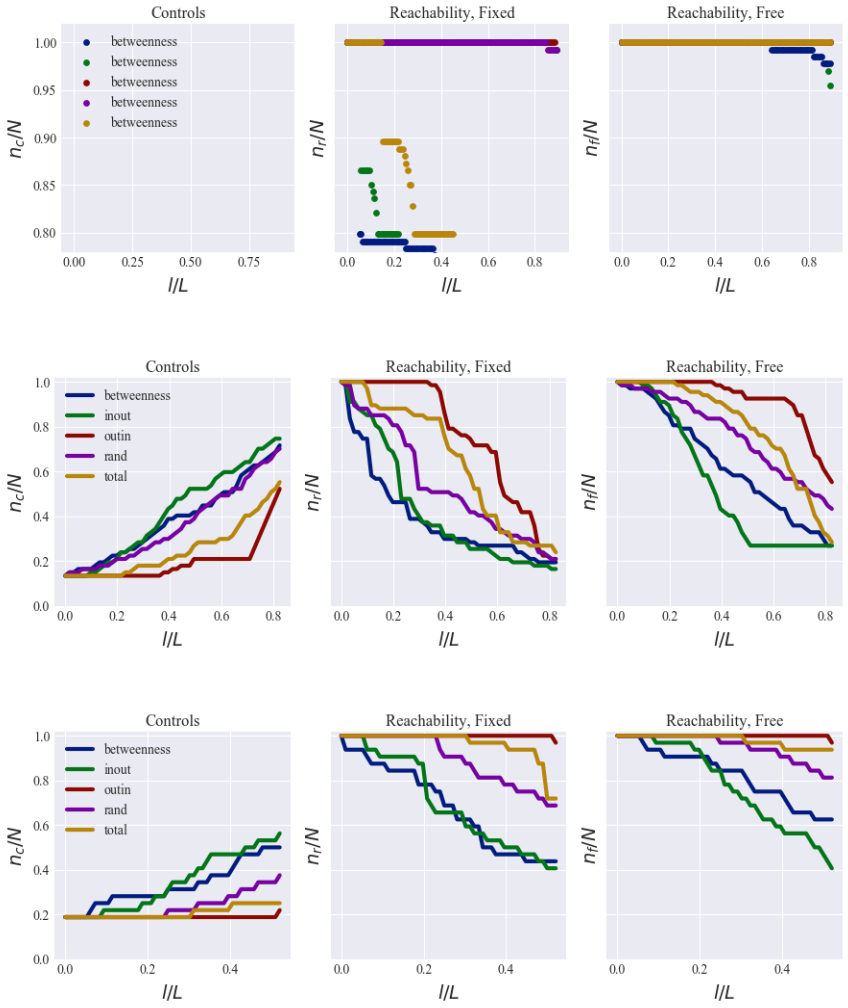



**Online Communication Graphs (top to bottom):** *dnc-emails*, *email-Eu-core*, *polblogs-reverse*, *one_mode_char*, *one_mode_message*.

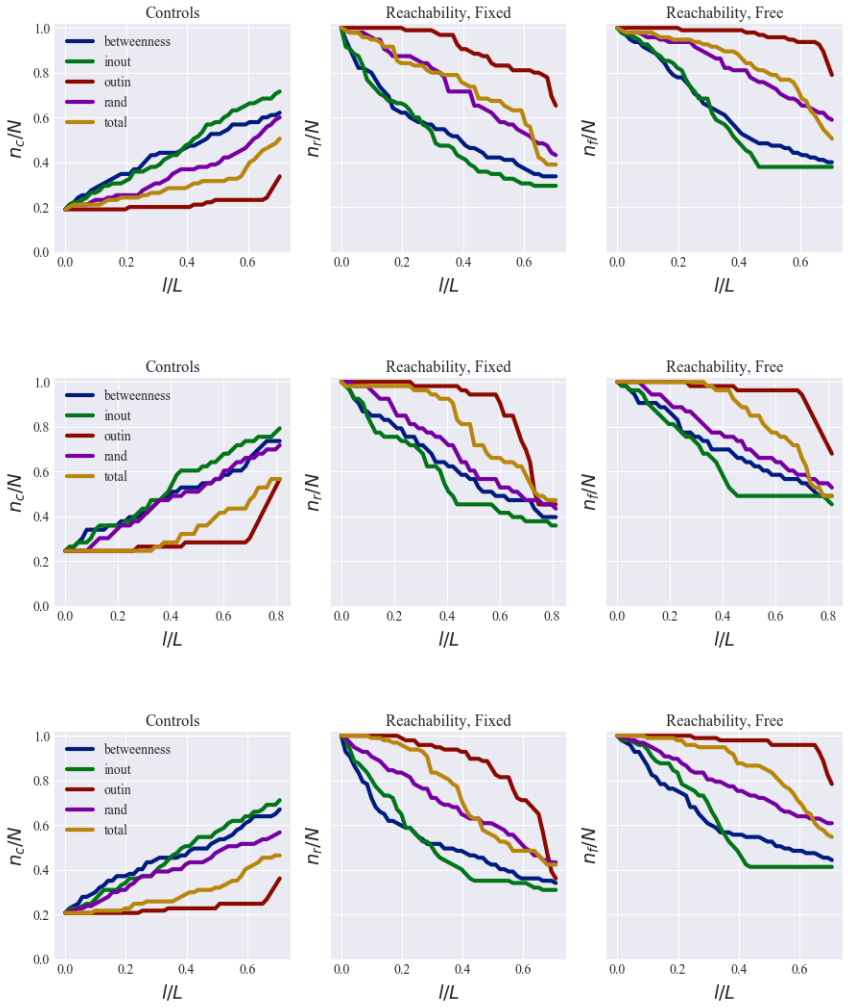



**Social Influence Graphs (top to bottom):** *physician-advice-reverse, physician-discuss-reverse, teacher-student*.

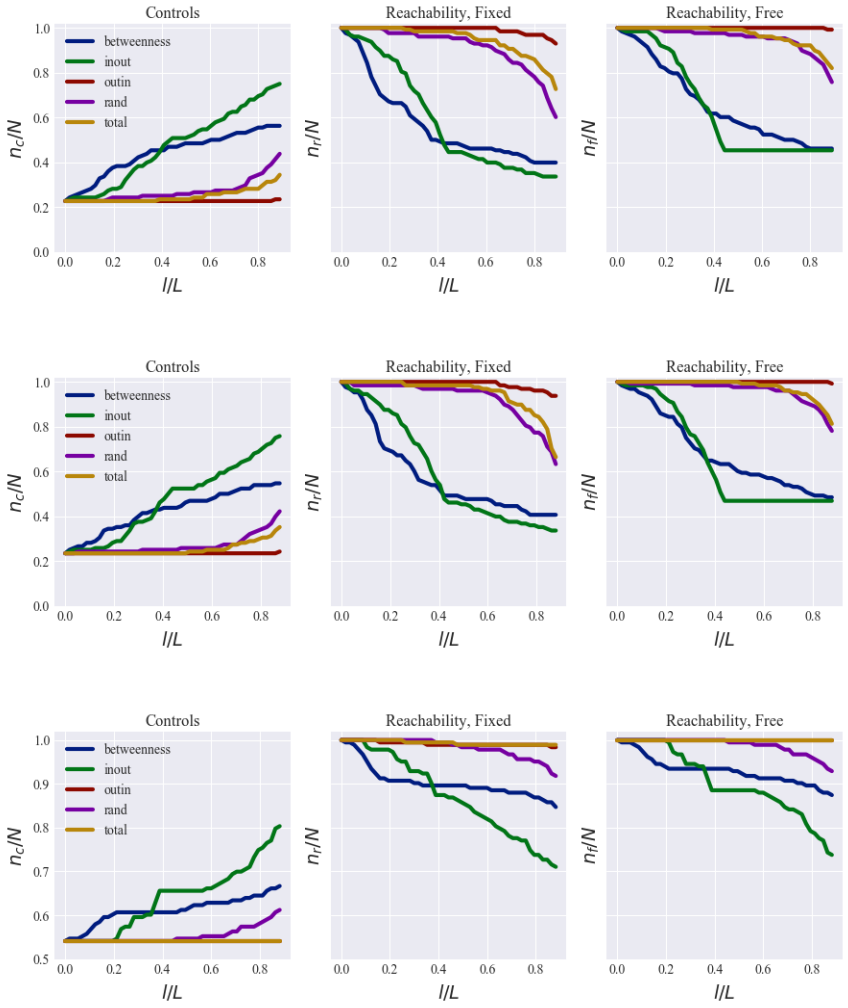

Controls
Reachability, Fixed
Reachability, Free
betweenness
inout
outin
rand
total
$n_c/N$
$n_{rf}/N$
$n_{ff}/N$
l/L



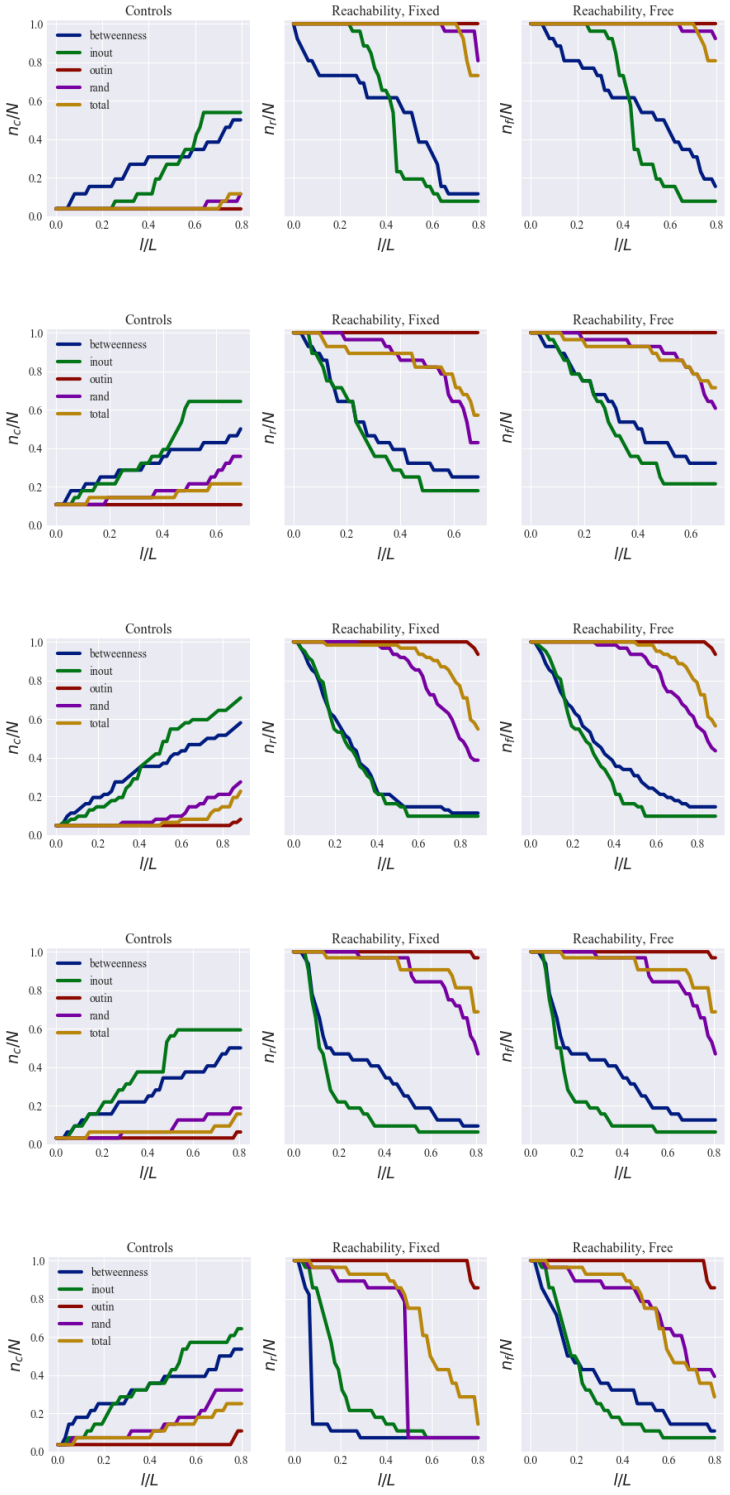

**Social Graphs (top to bottom):** *freeman1*, *freeman2*, *freeman3*, *physician-friend-reverse*, *prison*, *social1inter_st*, *social3inter_st*.



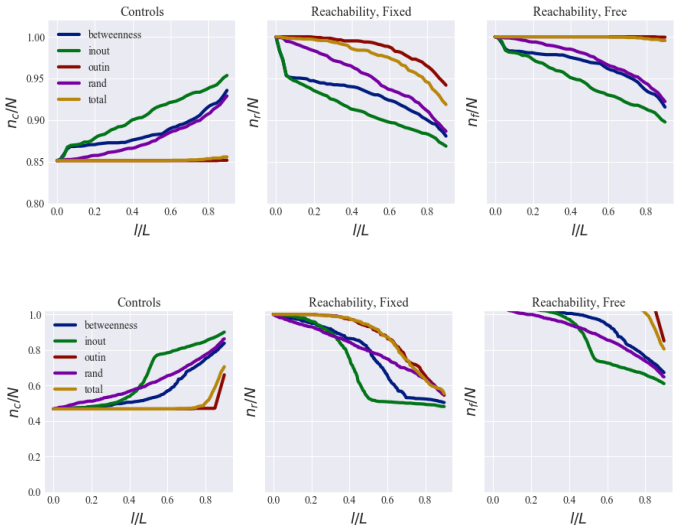

**Protein Structure Graphs (top to bottom):** *ps1*, *ps2*, *ps3*.



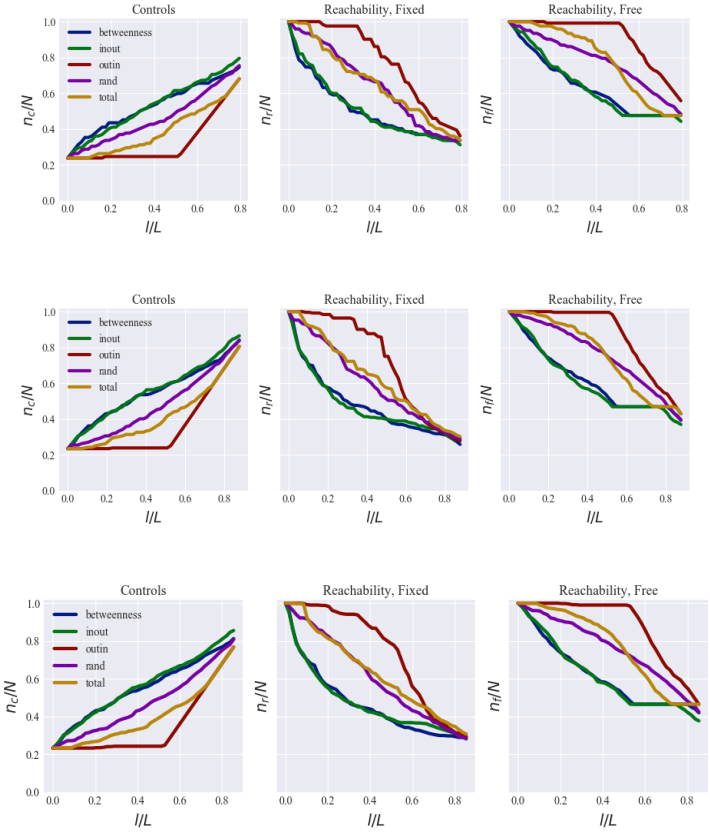

**Trophic Graphs (top to bottom):** *foodweb-baydry*, *foodweb-baywet*, *maayan-foodweb*.




**Animal Dominance Graphs (top to bottom):** *bison, cattle, macaque_dominance, sheep_dominance, hen_dominance.*




**Metabolic Graphs (top to bottom):** *figeys, protein_stelzl.*




**Electrical Graphs (top to bottom):** *s208, s420, s838.*

## APPENDIX B

## NETWORK DATA INFORMATION

**Infrastructure networks:** This includes a graph of US airports where nodes are the airports and edges are flights and the openflights network, a graph of two non-US based airports. Data and more information is found at: http://toreopsahl.com/datasets.

**C. Elegans:** This is a graph of the Caenorhabitic elegans worm's neural network. Neurons are the nodes and edges represent a synapse or gap junction between the neurons [8]. Data Source: http://toreopsahl.com/datasets.

**E-coli:** This network contains information describing the transcription regulation for E. coli encoding 577 interactions between 116 transcription factors and 419 operons [8]. Data Source: http://www.weizmann.ac.il/mcb/UriAlon/.

**Yeast:** This network describes the interaction in the yeast transcription network [8]. Data Source: http://www.weizmann.ac.il/mcb/UriAlon/.

**Macaque-Neural networks:** This is a collection of three networks describing the structural cortical connectivity in Macaque monkeys [8]. The data sources are: CoCoMac: http://cocomac.g-node.org/, Mac-95: https://sites.google.com/site/bctnet/datasets, Macaque-71: http://www.biological-networks.org/.

**Email-Eu:** The network is made from email data from a large European research institution [8]. Data Source: http://snap.stanford.edu/data/

**DNC Emails:** This network was made from email data from the Democratic National Committee. More information and data can be found here: http://konect.uni-koblenz.de/networks/.

**Physician networks:** These three networks describe relationships and interactions between physicians in four towns in Illinois [8]. Data Source: http://moreno.ss.uci.edu/data.html.

**Political Blog:** This network is created from political blog data on US politics in 2015 [8]. Data Source: http://www-personal.umich.edu/~mejn/netdata/

**Electronic Circuits networks:** These graphs describe electric circuits. Data Source: http://www.boseinst.ernet.in/soumen/Network_Controllability_Datasets.html

**Trophic networks:** These networks describe the animal food webs of the Florida ecosystem dry area and wet area, as well as in Little Rock Lake, Wisconsin. More information and data can be found here: http://konect.uni-koblenz.de/networks/.

**Animal Dominance networks:** These networks describe the dominance behavior of various animals. A node from the source to a target represents the dominance of the source over the target. More information and data can be found here: http://konect.uni-koblenz.de/networks/.

**Teacher-Student:** This networks described the relationship between teachers and students of the founding members of the International Network for Social Network Analysis. Data Source: http://moreno.ss.uci.edu/data.html.

**Metabolic networks:** These networks represent interacting pairs of human proteins. More information and data can be found here: http://konect.uni-koblenz.de/networks/.

**Intra-organizational networks:** These are four networks that describe relationships between employees of a consulting company and a research company [8]. Data Source: http://toreopsahl.com/datasets.

**Protein Structure networks:** Data Source: http://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks

**Social Interaction networks** (*social1inter, social3inter)*: The are social networks that capture positive sentiment. Data Source: http://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks

**Freeman networks:** These networks are made from data describing relationships between researchers. Data and more information is found at: http://toreopsahl.com/datasets.

**UC Irvine Messaging** (*one_mode_message, one_mode_char*)**:** These two networks are created from messaging data between UC Irvine students on a online platform similar to Facebook [8]. Data Source: http://toreopsahl.com/datasets**.**

## REFERENCES


[1] J. Thomas, Supratim Ghosh, Deven Parek, Derek Ruths, Justin Ruths, "Robustness of Network Controllability to Degree-Based Edge Attacks," *International Workshop on Complex Networks and their Applications,* pp. pp. 525-537, 2016.

[2] C. Stegehuis, R. van der Hofstad, and J. S. van Leeuwaarden, "Epidemic spreading on complex networks with community structures," *Scientific reports,* vol. 6, 2016.

[3] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences,* vol. 99, pp. 7821-7826, 2002.

[4] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E,* vol. 69, p. 026113, 2004.

[5] Barabasi. Available: http://networksciencebook.com/

[6] M. Newman, *Networks: an introduction*: Oxford university press, 2010.

[7] S. J. Banerjee and S. Roy, "Key to network controllability," *arXiv preprint arXiv:1209.3737,* 2012.

[8] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science,* vol. 343, pp. 1373-1376, 2014.

[9] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature,* vol. 473, pp. 167-173, 2011.

[10] P. J. Antsaklis and A. N. Michel, *A linear systems primer* vol. 1: Birkhäuser Boston, 2007.

[11] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control,* vol. 19, pp. 201-208, 1974.

[12] J. E. Hopcroft and R. M. Karp, "An n^5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing,* vol. 2, pp. 225-231, 1973.

[13] D. Parekh, D. Ruths, and J. Ruths, "Reachability-based robustness of network controllability under node and edge attacks," in *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*, 2014, pp. 424-431.

[14] C.-L. Pu, W.-J. Pei, and A. Michaelson, "Robustness analysis of network controllability," *Physica A: Statistical Mechanics and its Applications,* vol. 391, pp. 4420-4425, 2012.

[15] S. Nie, X. Wang, H. Zhang, Q. Li, and B. Wang, "Robustness of controllability for networks based on edge-attack," *PloS one,* vol. 9, p. e89066, 2014.

## BIOGRAPHICAL SKETCH

David Lanigan was born in Coatesville, PA and was homeschooled throughout high school. He moved to Texas to attend college and when he decided to pursue engineering as a career, transferred to The University of Texas at Dallas. He graduated with a BS in mechanical engineering, in December 2016 and remained at The University of Texas at Dallas to finish his master's degree, also in mechanical engineering.

## CURRICULUM VITAE

### David M. J. Lanigan

**Email**: dav.lanigan@gmail.com

**Education**: BS in Mechanical Engineering, UTD

**Research Interests**: Control Theory, Graph Theory, Complex Networks, Machine Learning, Robotics, Turbulence.

**Publications**: "Atmospheric stability and diurnal patterns of Aeolian saltation on the Llano Estacado" – *Aeolian Research*, 2016

**Conference Presentations**: "Low-cost robotic hand that senses temperature and pressure" - ASEE GSW 2017.

**Awards & Honors**: UTD Engineering and Computer Science Honors student, Phil Ritter Endowed Undergraduate Research Scholarship recipient, 2015.

**Professional Memberships**: ASME, SAE, Tau Beta Xi honors society