

MASTER

Improving network robustness using distance-based graph measures

Jurgens, S.E.

Award date:
2014

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Improving network robustness
using distance-based graph measures

Sander Jurgens

November 10, 2014



Master thesis

Department of Math and Computer Science
Eindhoven University of Technology

Thesis supervisors

dr. K.A. Buchin



Performance of Network and Systems
TNO ICT, Delft

dr. D.T.H. Worm

Abstract

Complex networks are ubiquitous in today's world, being the infrastructure of modern society. From power grids to transportation highways, network topologies can be found everywhere. As a result, their failure has major consequences on our daily lives, even more so if these problems cascade throughout the network. Therefore, improving the robustness of such networks becomes an essential task, making them able to better cope with occurring failures.

Robustness is defined as the ability of a network to continue performing well when it is subject to attacks and random failures. By representing networks as simple, connected, undirected, weighted graphs, we are able to quantify robustness with the use of some well known topological graph measures. These measures are based on distance, connectivity or centrality. Furthermore spectral measures are also considered.

In addition to surveying these graph measures through experimental evaluation, we also use the distance-based measures in the network robustness improvement process. This process aims at adding a limited number of edges to optimally improve the measures. Although the problem itself is NP-hard, our proposed approximation algorithms can still ensure a good solution. Finally, to demonstrate its applicability, a versatile software tool has been realized which is able to assist in analyzing and improving the robustness of any given network.

Contents

Abstract	3
List of Symbols	7
1 Introduction	9
1.1 Motivation	9
1.2 Related work	10
1.3 Objectives	11
1.4 Outline	11
2 Graph measures	13
2.1 Distance	13
2.2 Centrality	15
2.3 Connectivity	16
2.4 Spectral	19
3 Measure evaluation	23
3.1 Experimental setup	23
3.2 Measure comparison	28
4 Complexity of improvements	31
4.1 Covering by 3-sets	32
4.2 Eccentricity	33
4.3 Average Shortest Path	34
4.4 Radius	38
4.5 Diameter	40
5 Approximating optimal improvements	43
5.1 Clustering	44
5.2 Eccentricity	45
5.3 Average Shortest Path	50
5.4 Radius	53
5.5 Diameter	54
5.6 Algorithm evaluation	56
6 Conclusion	57
6.1 Discussion	57
6.2 Future research	58
Bibliography	64

List of Symbols

Graph theory

V, E	set of vertices, set of edges
n, m	size of V , size of E
$w(u, v)$	weight of edge (u, v)
$\delta(v), \delta_{min}$	degree of vertex v (number of adjacent vertices), minimum degree
$N(v)$	neighborhood of v (set of adjacent vertices)

Distance measures

$d(u, v), d_G(u, v)$	weighted shortest path distance between vertices u and v (in graph G)
$\epsilon(v), \epsilon_G(v)$	eccentricity of vertex v (in graph G)
R, R_G	radius (in graph G)
D, D_G	diameter (in graph G)
$D^{avg}(v), D_G^{avg}(v)$	single source average shortest path length for vertex v (in graph G)
D^{avg}, D_G^{avg}	average shortest path length (in graph G)

Centrality measures

$\bar{\delta}, \delta(v)$	(network average) degree centrality
$\bar{c}, c(v)$	(network average) closeness centrality
$\bar{b}_v, \bar{b}_e, b(x)$	(network average) vertex/edge betweenness centrality (of vertex/edge x)
$\sigma_{st}, \sigma_{st}(x)$	number of shortest paths between s and t (passing through vertex/edge x)

Connectivity measures

κ_v, κ_e	vertex connectivity, edge connectivity
ν	largest connected component size
$\bar{\rho}, \rho(v)$	(network average) clustering coefficient

Spectral measures

A	adjacency matrix
Δ	degree matrix
L, L^W	Laplacian, weighted Laplacian
λ, λ^W	vector of (weighted) Laplacian eigenvalues
λ_i, λ_i^W	the i th smallest (weighted) Laplacian eigenvalue
λ_2	algebraic connectivity
Ω	effective graph resistance

Improvement problems

x3C, 3C	(exact) cover by 3-sets
U-EM, EM	(unit weight) eccentricity minimization
U-RM, RM	(unit weight) radius minimization
U-DM, DM	(unit weight) diameter minimization
U-SSASPM, SSASPM	(unit weight) single source average shortest path length minimization
U-ASPM, ASPM	(unit weight) average shortest path length minimization

Approximating improvements

ω	weight of added edges
----------	-----------------------

Chapter 1

Introduction

1.1 Motivation

The world we live in nowadays is highly connected. Interaction with surrounding networks occurs on a continuous basis. Therefore, the consequences for failure have never been higher. They range from large-scale power blackouts to major traffic congestions, or even a complete telecommunication breakdown. Although these problems may seem like a temporary nuisance, they have the ability to create serious issues for vital services.

An example of a network that is susceptible to failure is the U.S. airline system. This network is based on a model where cities are linked together through highly connected hubs, as shown in red in Figure 1.1. Attacking these hubs could easily disconnect the network, bringing air traffic between the two coasts to a halt.

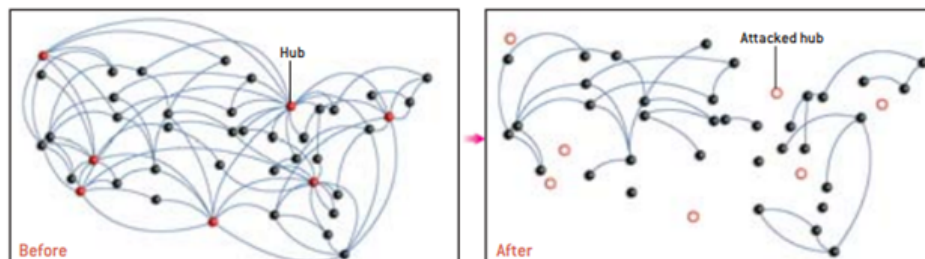


Figure 1.1: Simplified U.S. airline network before and after attacking the hubs.

Robustness of networks, in general, is of high importance due to the societal risks involved. Moreover, many real world networks are the result of different processes that may not have taken robustness into account, making it increasingly important to study their robustness at this time. To introduce the concept of robustness a definition of the term is given.

Definition 1.1. *Robustness* is the ability of a network to continue performing well when it is subject to attacks and random failures.

An approach with accurate measurements is needed in order to quantify network robustness and to check whether a network is actually as robust as it seems. This quantitative approach would need to measure the network connectivity, preventing failures from having a large impact by taking down connections. To this extent many measures have been proposed in

the literature. In this thesis we will be going back to the fundamentals of graph theory to look for simple and intuitive graph measures that give an indication of robustness.

Once the measures have been established, we can make use of them to improve the robustness of existing networks. Additionally, it can provide guidance in the design of new, more robust, networks. However, costs are often involved for network modification. Hence, we will limit ourselves in the number of modifications, whilst still providing close to optimal improvements in robustness. By doing so, we make the proposed approach viable for real life scenarios.

1.2 Related work

A large number of networks we use every day are extremely vulnerable due to their simplistic structure, which is why robustness studies are prevalent in network research. These susceptible networks are often based on a scale-free model, the degree distribution of such models allows them to be easily expandable. This type of network is also beneficial with respect to maintenance, since it can mostly be focused on the highly connected hubs. However, these hubs also present a risk to robustness, being very sensitive to failures and targeted attacks [3].

Although many studies have been done on scale-free networks, robustness research spans across all types of networks and disciplines. This ranges from modeling power networks [25] to area failures, e.g. floods, in telecom networks [1], and even the spread of disease and other ecological problems [38]. In the end, all these examples come down to robustness analysis of the underlying network topology, which is why this thesis focuses purely on graph theory in a mathematical context.

Complex approaches to robustness analysis are quite common in the literature, since robustness itself is very multifaceted. In recent studies researchers have looked at the existence of alternative paths as a measure for robustness [44]. Others claim dynamic techniques work better, such as analyzing cascading failures [45] and back-up routing possibilities [36]. The amount of usable measures is so staggeringly high that frameworks have even been proposed to provide guidelines and generalizations for new measures [29, 30, 39].

We believe it is time to go back to basics by using simple topological graph measure in an attempt to improve the robustness of any network. This improvement will be done by edge addition. Although simple randomized edge addition schemes already exist [6], more sophisticated approaches are required such as the ones aimed at diameter reduction [8, 10, 16, 23]. Techniques like these have sparsely been applied to different measures, for example minimizing the average shortest path lengths with zero-weight shortcuts [32], but a general approach does not yet exist.

Overall, when improving robustness, distance-based techniques seem to be more common in literature. However, improvement based on different measures also exists, namely in the field of spectral graph theory as extensively discussed by Cvetković et al [9]. Examples from literature include algebraic connectivity improvement through linear algebra methods [41] and optimization of the effective resistance based on a statistical approach [42].

In addition to robustness improvement, some work has also been done in network simplification through edge deletion. The goal behind these approaches is to reduce operational costs with the lowest sacrifice in robustness. Complexity proofs for such problems were done by Schoone et al [35], whereas other techniques using minimum spanning trees were also introduced [20]. In this thesis we will only consider edge addition.

1.3 Objectives

The aim of this thesis is to find intuitively clear network robustness measures and utilize them in the process of robustness improvement. To this extent we will represent networks as simple, connected, undirected, weighted graphs, quantifying robustness in terms of fundamental topological graph measures. There is no loss of generality in this representation, meaning that the approach is applicable to real-world scenarios. Also, to remain close to real life, we limited ourselves in the number of allowed modification which complicates the process significantly. This results in the following research question.

Research question *How can we determine the robustness of a simple, connected, undirected, weighted graph, and use the resulting measures in the improvement process?*

We will answer this question by taking an in depth look at several graph measures, performing comparisons between them to show their effectiveness. As for improving the robustness, we will take the distance-based graph measures as a basis and prove the complexity of the improvement when limiting ourselves in the number of modifications. Furthermore, we provide approximation algorithms for computing improvement suggestions whose resulting measure value is within a certain bound from the optimal. This bound is also formally proven. In short, we intend to answer the following questions:

Secondary research questions

- Which graph measures are suited for measuring network robustness?
- How viable are distance-based measures in measuring the network robustness?
- What complexity class does robustness improvement, i.e. distance-based measure minimization, belong to?
- How do we go about approximating the optimal robustness improvements?
- What approximation ratios can we achieve for robustness improvements?

1.4 Outline

This thesis will go through the research questions in order, starting with Chapter 2 which introduces a broad range of graph measures. Each section in this chapter discusses a class of topological graph measures based on a specific aspect of graph theory: distance, centrality, connectivity and spectral respectively. A short intuitive explanation regarding the relevance of each measure to the network robustness is also given in each section.

In Chapter 3 we expand on this explanation of relevancy by experimentally validating the capability of each graph measures to capture network robustness. The central question is whether these measures, which are not specifically introduced as network robustness measures, could be used to determine the robustness properties of a graph. To this end a versatile software tool which we developed is introduced in Section 3.1 providing the experimental setup, whereas the actual results are processed in Section 3.2.

Assuming the distance-based measures to be viable in capturing network robustness, we are able to improve network robustness by adding edges. However, optimally minimizing these measures with a limited supply of additional edges is NP-hard as proven in Chapter 4. In our hardness proofs we will reduce a known NP-complete problem, a variant on set cover as defined in Section 4.1, to these minimizations. All other sections in this chapter present a variant of this reduction for a specific distance-based measure.

Finally, Chapter 5 is dedicated to the minimization of the distance-based graph measures through the addition of a set amount of edges. Although this problem was proven to be NP-hard in the previous chapter, algorithms are designed to approximate the optimal solution to this minimization within a proven bound. All of these algorithms rely on a clustering technique that is introduced in Section 5.1. The other sections in this chapter are focused at minimizing specific distance-based measures.

Chapter 2

Graph measures

The networks considered in this thesis can be represented by graphs, which allows us to utilize the numerous measures that have been introduced to characterize them. We define such a graph $G = (V, E)$ to be simple, undirected, connected and weighted, with $|V| = n$ vertices and $|E| = m$ edges. The edge weights are represented by function w , mapping pairs of vertices to a positive real number.

$$w : V \times V \rightarrow \mathbb{R}_0^+$$

For unweighted graphs we assume edges to have unit weight, i.e. $w(u, v) = 1$ if there is an edge between $u, v \in V$.

In this chapter we summarize some of the more well known topological graph measures, whose relevancy in robustness measurements is quite intuitive. Each section describes a class of measures, and discusses how they effectively measure network robustness. An experimental evaluation of these measures is performed in the next chapter.

2.1 Distance

Distances, both between individual vertices as well as in the graph overall, present an easy and intuitive way of defining robustness. These distances are defined on a path, which is a sequence of vertices $\pi = (v_1, \dots, v_k) \in V \times \dots \times V$ such that $(v_i, v_{i+1}) \in E$ for $1 \leq i < k$. As is expected, shorter paths are less vulnerable, since this gives an indication that the vertices are much more closely connected. Longer paths on the hand show a lower robustness value, because the probability of failure increases with each additional edge on the path.

We define the (weighted) length of a path (v_1, \dots, v_k) to be $\sum_{i=1}^{k-1} w(v_i, v_{i+1})$, i.e., the combined weight of all edges that make up the shortest path. The shortest path between vertices $u \in V$ and $v \in V$ is a path with minimal length, hence no other path has a smaller length. This path need not be unique. Let distance function $d(u, v)$ be the (weighted) length of the shortest path between u and v . Similar to the edge weights, it maps all pairs of vertices to a positive real number

$$d : V \times V \rightarrow \mathbb{R}_0^+$$

Summarizing, we define the distance function as follows.

Definition 2.1. $d(u, v)$ is the combined weight of all edges that make up the shortest path, with respect to their edge weights, between vertex $u \in V$ and vertex $v \in V$.

Using this distance function we can define multiple distance-based measures, starting with *eccentricity*. Eccentricity is, in essence, a distance-based measure between individual vertices, but also a helpful concept in explaining some other measures in this section. For a vertex $v \in V$ it represents the greatest geodesic distance between v and some other vertex in the graph. Formally, it is defined as follows.

Definition 2.2. *Eccentricity*, for a vertex $v \in V$, is given by

$$\epsilon(v) = \max_{u \in V} d(v, u).$$

The concept of eccentricity can be applied in specifying the *radius* measure. As its name suggests, it describes the radius of a graph as seen from a *central vertex*. In a graph with radius R a central vertex is one whose eccentricity is exactly R , meaning all other vertices are reachable within that distance. This leads to the following definition for the radius measure.

Definition 2.3. The *radius* of a graph is the minimum eccentricity over all of its vertices:

$$R = \min_{v \in V} \epsilon(v).$$

An intuitive, and more well known, counterpart to the radius is the *diameter* measure. The diameter got its name from the fact that it describes the largest distance between any two vertices in the graph, more specifically any two *peripheral vertices*. Given a graph with diameter D a peripheral vertex is one that defines the diameter, hence whose eccentricity is exactly D . This measure is defined as follows.

Definition 2.4. The *diameter* of a graph is the maximum eccentricity over all of its vertices:

$$D = \max_{v \in V} \epsilon(v).$$

All of the measures thus far deal with extreme values for the distances, this has become very prominent in literature due to the relative easy of reasoning about these values. However, an average value, although more difficult to theoretically analyze, says more about the network as a whole in comparison to the radius and diameter which deal with more extreme cases. This is the reason why we include the *average shortest path* (ASP) measure, both the single source as well as the general variation, which averages the shortest paths between all combinations of vertices in the graph.

Definition 2.5. The *single source average shortest path* length of a vertex is the average over the shortest paths to all other vertices in the graph:

$$D^{avg}(v) = \frac{1}{n-1} \sum_{u \in V} d(u, v)$$

Definition 2.6. The *average shortest path* length of a graph is the average over the shortest paths between all combinations of vertices:

$$D^{avg} = \frac{1}{n} \sum_{u \in V} D^{avg}(u) = \frac{1}{n(n-1)} \sum_{u \in V} \sum_{v \in V} d(u, v).$$

Overall, these distance-based measures give a good insight as to how closely the graph is connected, which is a sign of robustness against random failures. Some measures are more well known than others, but all provide the ability to analyze specific parts of network robustness. Averaged distance measures such as ASP are more sensible than ones which rely on extreme values, since they tell more about the graph as a whole. On the other hand, the measures based on extreme values tell us more about worst case performance.

2.2 Centrality

In graph theory, centrality aims at identifying the most important vertices within a graph, ones that are most central to its structure. Important vertices are characterized by their topological location, large number of connections, and the amount of information passing through. These measures find their origin in social networks, where it is able to identify the most influential person(s) [34]. Other applications include locating key infrastructure vertices of urban networks and finding spreaders of disease.

Since most centrality measures only govern the importance of single vertices, and not network robustness as a whole, they are not as important to us as the other measures presented in this chapter. However, they do show some interesting concepts which we will be able to use later on, such as locating interesting graph elements to attack, which is why they are still a point of discussion in this thesis.

Historically the first centrality measure, and arguable the simplest graph measure overall, is the *degree centrality*, which is defined as the number edges incident upon a vertex. In terms of robustness, the degree can be interpreted as the failure risk of a vertex, due to its utilization in the network. More neighbors means a higher probability of failure, and hence a lower robustness. An example of these types of failures can be found in communication networks, where servers are often overloaded with many incoming service requests (i.e. a high degree).

Definition 2.7. *Degree centrality* of a vertex $v \in V$ is the number of edges incident upon said vertex:

$$\delta(v) = |\{ u \in V \mid (u, v) \in E \}|.$$

As specified at the beginning of this chapter, we are dealing with undirected graphs. Hence we will not distinguish between in-degree and out-degree. An averaged version of this measure, capable of measuring overall network robustness, is defined as follows.

$$\bar{\delta} = \frac{1}{n} \sum_{v \in V} \delta(v)$$

Centrality can also be based on distances to (all) other vertices, as is done with the *closeness centrality* measure. Closeness is defined as the sum of the inverted distances, i.e. farness, to all other vertices in the graph. Thus, more central vertices have a lower closeness, since they are closer to all other vertices. Intuitively, closeness can be regarded as a measure of how long it will take to spread information from to all other nodes sequentially.

Definition 2.8. *Closeness centrality* of a vertex $v \in V$ is the sum of inverse distances to all other vertices:

$$c(v) = \sum_{u \in V \wedge u \neq v} \frac{1}{d(u, v)}.$$

To measure overall robustness, and more specifically whether there is a high percentage of outliers, a variation of this metric averaged over all vertices is introduced. This variation, which is quite similar to the average shortest path measure, can be computed as follows.

$$\bar{c} = \frac{1}{n} \sum_{v \in V} c(v)$$

The final centrality measure which we will introduce is *betweenness centrality*. This measure is special in that it is applicable to both vertices and edges, making it ideal for selecting important graph elements to attack. Betweenness centrality denotes the number of shortest paths, when considering all combinatorial pairs of vertices, passing through a vertex or edge. In terms of robustness measurements, this means that the measure excels in networks with information traveling along the edges.

Definition 2.9. *Betweenness centrality* of a vertex or edge x is the number of shortest paths passing through it:

$$b(x) = \sum_{s,t \in V \wedge s \neq x \neq t} \frac{\sigma_{st}(x)}{\sigma_{st}}.$$

where σ_{st} is total number of shortest paths from vertex s to vertex t and $\sigma_{st}(x)$ is the number of those paths that pass through x .

If desired, the betweenness can be normalized by dividing through the number of vertex pairs excluding x , which is $(n-1)(n-2)/2$ in our graphs. As for the overall robustness of a network, Sydney et al have proposed a measure based on maximum edge betweenness and its behavior as vertices are removed [37]. However, as we are interested in comparisons with other averaged centrality measures, we introduce an averaged variation of the betweenness centrality measure below, both for vertex and edge betweenness.

$$\bar{b}_v = \frac{1}{n} \sum_{v \in V} b(v) \quad \bar{b}_e = \frac{1}{m} \sum_{e \in E} b(e)$$

2.3 Connectivity

Connectivity is one of the basic concepts of graph theory, it implies the existence of connections between components of the graph. This means that a highly connected graph is more dense, whereas a graph with less connectivity is sparse. In terms of robustness, a dense graph is more resilient, having back-up paths if anything goes wrong. Therefore, connectivity-based graph measures are also intuitively compatible as robustness measures.

An important part of connectivity is how well the network is able to hold together. Hence, when discussing connectivity, the classic measures *vertex connectivity* and *edge connectivity* first come to mind. These measures express the amount of removals it takes to disconnect the graph into multiple separate connected components. More formally, they are defined as follows.

Definition 2.10. The *vertex connectivity* of a graph is the minimum number of vertices κ_v whose deletion from the graph disconnects it into multiple components.

Definition 2.11. The *edge connectivity* of a graph is the minimum number of edges κ_e whose deletion from the graph disconnects it into multiple components.

The act of disconnecting a graph into multiple connected components can be seen in more detail in Figure 2.1. The figure shows the removal of a single graph element, denoted by the vertex and dotted edges, which leads to the graph splitting into two components.

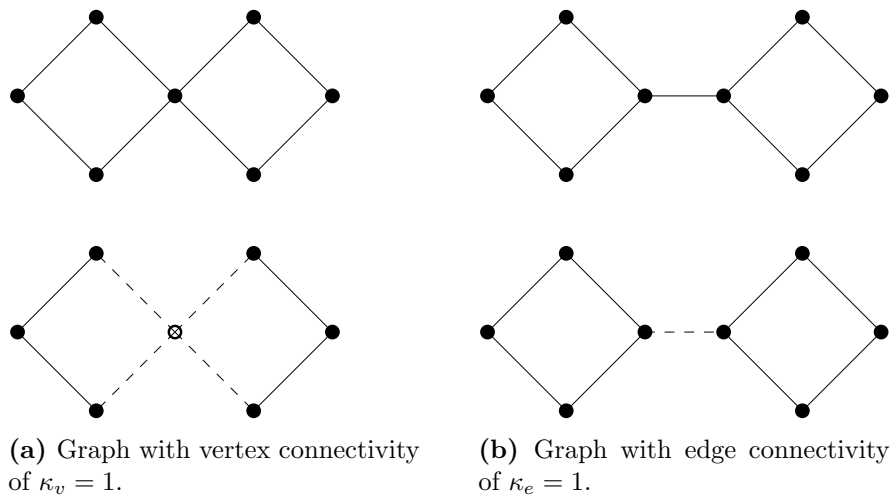


Figure 2.1: Disconnecting a graph into multiple components.

It should be noted that the vertex and edge connectivity have a special property between them. Since deleting a vertex also influences the connected edges, it holds that $\kappa_v \leq \kappa_e \leq \delta_{min}$, where δ_{min} is the minimum degree of all vertices in the graph.

Also, vertex connectivity has an interesting case for a complete graph, which by definition cannot be disconnected by vertex deletions. To uphold the given inequality we define the vertex connectivity to be $\kappa_v = n - 1$ for this case.

When looking at connectivity in a more local setting, we are in need of different measures to capture these specific details of robustness. *Clustering coefficient* is such a measure, it determines connectedness through the presence of triangles between neighboring vertices and comparing it to the total number of connected triples. Originally this measure was designed for social networks, where the context is the probability that two friends of a person are each others friends too, but it can also be used to measure robustness.

In terms of robustness, a better clustering is inherent to a higher robustness due to the increasing number of alternative paths for each existing triangle. However, before we can continue to describe the clustering coefficient, we must first define the neighborhood of a vertex which enables the creation of triangles in the graph.

Definition 2.12. The *neighborhood* of a vertex consists of the immediately connected vertices of the graph:

$$N(v) = \{ u \in V \mid (u, v) \in E \}.$$

Using this definition for the neighborhood of a vertex, we can more formally describe the clustering coefficient. As shortly explained above, the clustering coefficient is computed by dividing the number of triangles, also called closed triples, by the total number of connected triples. In an undirected graph this can be defined as follows:

Definition 2.13. The *local clustering coefficient* of a vertex $v \in V$ is the number of triangles containing v divided by the total number of connected triples containing v :

$$\rho(v) = \frac{2 |\{ (u, w) \mid u, w \in N(v), (u, w) \in E \}|}{\delta(v)(\delta(v) - 1)}.$$

An undirected graph has the property that (u, v) and (v, u) are considered identical. Therefore, if a vertex v has $\delta(v)$ neighbors, $\delta(v)(\delta(v) - 1)/2$ edges could exist among the vertices within the neighborhood. In a directed graph this amount would be $\delta(v)(\delta(v) - 1)$. For additional clarity, the type of triples this measure tries to locate are visualized in Figure 2.2.

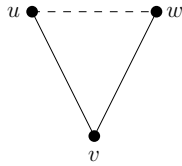


Figure 2.2: Vertex v has neighbors u and w which may or may not form a closed triple (triangle).

As expected, the global clustering coefficient is defined in the same way as its local variant; dividing the total number of closed triples by total number of connected triples of vertices. However, due to the complexity of computing this measure, the network average clustering coefficient is often considered instead as suggested by Watts and Strogatz [43]:

$$\bar{\rho} = \frac{1}{n} \sum_{v \in V} \rho(v)$$

Although this measure captures the existence of triples very well, which is what we are mainly interested in, it does not take edge weights into account. Therefore, a generalization of this measure which applies to weighted networks has been proposed by Barrat et al [5].

A somewhat simpler measure, yet still extremely relevant to connected components and connectivity in general, is the *largest connected component size*. Often named the *giant component size*, this measure aims at describing the size of the remaining active network. In the majority of real world cases the largest component contains most vertices critical to survival of the network, e.g. generators and other types of servers. Hence, all other smaller components will not survive independently.

Definition 2.14. The *largest connected component size* of a graph is the number of vertices, ν , present in the largest connected component of said graph.

Note that this measure plays a smaller role in connected graphs, which only have a single connected component, whereas its importance increases once failures or attacks disconnect the graph into multiple components.

Overall, it seems natural to say that connectivity is inherent to robustness. On the other hand, a graph split into multiple connected components might still be functioning well if there was already little communication between these components in the first place. Therefore, although the measures might seem intuitive, caveats can always be found.

2.4 Spectral

In all of the previous sections in this chapter we have studied graphs directly, extracting measures that are evident when representing the graph visually. However, different approaches exist, such as studying the matrices representing the graph. Presented in this section are several robustness measures dependent on the eigenvalues of the adjacency or Laplacian matrices. Our goal here is not to give all details about these measures, nor will we prove any of its properties, we will merely present the measures themselves. However, before we are able to introduce these measures, we must first define some of the basics. A more general introduction to spectral graph theory was written by Cvetković et al [9].

Definition 2.15. For all indices $1 \leq i, j \leq n$, where the graph contains vertices $v_i, v_j \in V$, the *adjacency matrix* A has at position (i, j) a one when edge (v_i, v_j) is present in the graph and a zero otherwise:

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.16. For all indices $1 \leq i \leq n$, where the graph contains vertices $v_i \in V$, the *degree matrix* Δ has at position (i, i) the degree of vertex v_i and 0 elsewhere:

$$\Delta_{ij} = \begin{cases} \delta(v_i) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

By combining the two matrices defined above we are able to create the *Laplacian matrix*, also called *admittance matrix*, which fully characterizes the original graph; when given, the entire graph can be reconstructed. The Laplacian matrix can in turn be used to find many other graph properties in the field of spectral graph theory.

Definition 2.17. The *Laplacian matrix* L is the difference $\Delta - A$ between the *degree matrix* Δ and the adjacency matrix A :

$$L_{ij} = \begin{cases} \delta(v_i) & \text{if } i = j, \\ -1 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

The most important property of the Laplacian matrix, with respect to robustness, is its second smallest eigenvalue. Often called the *algebraic connectivity*, this eigenvalue measures the connectivity of the graph, more specifically the difficulty to cut the graph into multiple connected components.

Definition 2.18. The *algebraic connectivity* λ_2 is the second smallest eigenvalue in the eigen-decomposition of the Laplacian matrix

$$L\mathbf{v} = \lambda_2\mathbf{v}$$

where L is the Laplacian matrix and \mathbf{v} a non-zero eigenvector, also called the Fiedler vector for λ_2 .

The algebraic connectivity, as originally proposed by Miroslav Fiedler, is a spectral measure which is closely related to a lot of other graph characteristics [14]. Some interesting properties of these characteristics are noted below, though a complete survey of its properties and applications was done by Mohar et al [33].

As for the properties, since we are dealing with undirected graphs, the adjacency matrix is symmetric and thus has real eigenvalues. This is very important because it allows for easy comparison with other robustness measures. Also, eigenvalue λ_2 starts at 0 when the graph is disconnected, and is limited by the vertex connectivity in graphs that are not complete. Hence the following inequality $0 \leq \lambda_2 \leq \kappa_v \leq \kappa_e \leq \delta_{min}$ holds for graphs that are not complete, as proven by Fiedler [14].

However, criticism is also to be found, as Baras and Hovareshti have shown that the algebraic connectivity does not always increase whenever an edge is added [4]. Figure 2.3 shows this example in more detail.

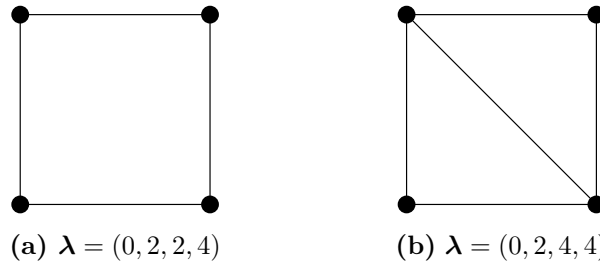


Figure 2.3: Two graph with the same algebraic connectivity.

Spectral graph theory also allows the use of edge weights in the matrices, this is done by slightly altering the definition of the Laplacian matrix to form a *weighted Laplacian matrix*. Though it should be noted that non-negative edge weights are not allowed in this matrix configuration.

Definition 2.19. The *weighted Laplacian matrix* L^W is the difference between the *strength diagonal*, where each element is the sum over all its connected edge weights, and the *weight matrix*:

$$L_{ij}^W = \begin{cases} \sum_{v_j \in V} w(v_i, v_j) & \text{if } i = j, \\ -w(v_i, v_j) & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the strength diagonal is introduced to keep the row sums at 0, similar to the unweighted Laplacian L . This is done to keep the all-one vector as an eigenvector with an eigenvalue of 0.

A spectral graph measure that uses this weighted Laplacian matrix extensively is the *effective graph resistance*. This measure aims at describing the global resistance of a graph as if all its edges were connected in parallel, where the edge weights correspond to its conductance (i.e. inverse of resistance). Hence, a graph with a relatively small effective resistance is implied to be more robust, due to less flow resistance along its edges.

Although this measure can be computed by summing the effective resistance of each separate edges in the graph, we will only present the more efficient global approach. An elaborate explanation of both approaches has been written by Ellens [12] and Koç et al [25, 26, 27].

Definition 2.20. The *effective graph resistance* Ω is the aggregated resistance as computed by the eigenvalues of the weighted Laplacian matrix:

$$\Omega = n \sum_{i=2}^n \frac{1}{\lambda_i^W}$$

where λ_i^W is the i th smallest eigenvalues of L^W .

Overall both presented spectral graph measures show potential. Algebraic connectivity has already established its position as a well known spectral graph measure with respect to robustness, whereas effective graph resistance is still relatively new. However, in contrast to algebraic connectivity, the effective graph resistance has the desirable property that its value strictly decreases when edges are added.

Chapter 3

Measure evaluation

A large number of graph measures has been introduced in Chapter 2. However, before we can use these measures in our analysis, experimental evaluation of their effectiveness with respect to capturing the robustness of any given network still needs to take place. This evaluation will analyze each graph measure on a number of different network topologies, both predefined (e.g. cycle, line, star) and randomly generated (e.g. scale-free, small-world).

Although this analysis could be done in any mathematical processing application, there is a demand for a customizable network analysis tool that is able to run on multiple platforms. This demand exists both in the consultancy branch where advice is given about network upgrades, and the companies who maintain the networks themselves. To this end, we developed a tool that is able to apply all measures mentioned in this thesis to any given network, and also contains algorithms to improve the network robustness as will be detailed later.

We will start this chapter by describing all functionality of the tool in Section 3.1, having subsections for each stage of the analysis process. Afterwards, in Section 3.2, the results that were gathered from the experimental validation are presented. Hence, we are able to conclude which measures are good at capturing network robustness.

3.1 Experimental setup

The developed Java tool, being able to run on a large number of platforms, has a workflow that allows users to generate, visualize, simulate, analyze and improve any given network. Each of these aspects is discussed in their respective subsections below. However, first we will define the network model that the tool is working with, since it provides a generalization on the standard definition of graph as presented in Chapter 2.

Network model

An *Interdependent Flow Network*, or IFN, is a network model with support for both interdependencies and flow computations. This specialized network model will only be relevant in Section 3.1 due to its presence in the tool. It will not be referred to in other parts of this thesis. The tool supports this model in particular since it is very adaptable to real life situations where flow and network interdependency are common occurrences.

We define an interdependent flow network as a sequence of n graphs, $\mathcal{G} = (G_1, G_2, \dots, G_n)$, where each graph is a tuple, $G_i = (V_i, E_i)$, consisting of its vertices (V_i) and edges (E_i). Vertices from graph G_i will be denote as either $u \in V_i$ or $u_i \in V_i$, where identifier i may be

added to a vertex to determine its graph of origin.

The interdependence between graphs is given by set \mathcal{I} , which contains all dependencies. A dependency occurring in the set can only exist between two different graphs, meaning that for a given dependency $(u_i, v_j) \in \mathcal{I}$, where $u_i \in V_i$, $v_j \in V_j$, and $1 \leq i, j \leq n$, it must hold that $i \neq j$. Dependencies can be directed or undirected, where a directed interdependency $(u_i, v_j) \in \mathcal{I}$ is defined as u_i depends on v_j . This results in u_i failing whenever v_j fails, but not the other way around.

Although flow computation on the network are supported, its representation with respect to the model will not be formally defined here. However, we will describe that each vertex in the IFN model can have a supply and demand, which means that it can be both a source and a sink at the same time. Each graph stores its sources and sinks in separate sets. Also, edges have flows and capacities as is normal in a flow network. Constraints regarding capacities, flow conservation and demand fulfillment are not enforced by the tool, but rather by the applied network simulators.

Generating networks

Although the tool provides the capability to open user-defined networks, generating random graphs is also part of validating the graph measures presented in Chapter 2. Hence, we implemented some of the most well known models:

- Erdős-Rényi models (both variants) [13]
- Barabási-Albert (scale-free) [2]
- Watts-Strogatz (small-world) [43]

The two closely related variants of the Erdős-Rényi random graph model supported in this tool are:

- The $G(n, M)$ model, where a graph is chosen uniformly at random from the collection of all graphs which have n nodes and M edges.
- The $G(n, p)$ model, where a graph is constructed by connecting nodes randomly. Each possible edge is included in the graph with probability p . Hence, for a given M all graphs with n nodes and M edges have equal possibility to be generated.

On the other hand, the Barabási-Albert model presents a different approach to generating random graphs. Its aim is to create scalable graphs whose degree distribution follows a power law. Many real life networks are conjectured to be scale-free which increases the applicability of the model in our tool.

The final random graph model is Watts-Strogatz, known for its small-world property. This property implies that most vertices are not neighbors of one another, but rather can be reached by a small number of steps. Moreover, the distance between two randomly chosen vertices grows proportionally to the logarithm of the number of vertices in the network. Such a model is most commonly used to emulate social networks.

Visualization

To visualize interdependent flow networks we used the *Java Universal Network/Graph Framework* (JUNG). This open-source software library provides a common and extendible language for the modeling and visualization of data that can be represented as a graph or network. By extending their definition of a graph with the IFN model, as previously presented, we are able to utilize the visualization framework to interactively explore different types of network data.

Although the tool is able to visualize networks based on its vertex locations, as given by the IFN model, the JUNG framework also comes with a number of graph layout algorithms built-in. Most of these are based on a force-directed approach, but some other techniques exist such as the competitive learning strategy used in the ISOM algorithm. The following list of layouts is implemented by the tool, some of which are depicted in Figure 3.1.

- Fruchterman-Reingold [17]
- Kamada-Kawai [22]
- Spring embedders [11]
- Self-organizing graph/map (ISOM) [31]

Additionally, two custom layout algorithms based on network interdependency clustering are implemented. These layouts cluster either networks or interdependent vertices into separate groups, each of which has a custom layout. The layout algorithms were added to provide a clear overview of existing interdependencies between networks.

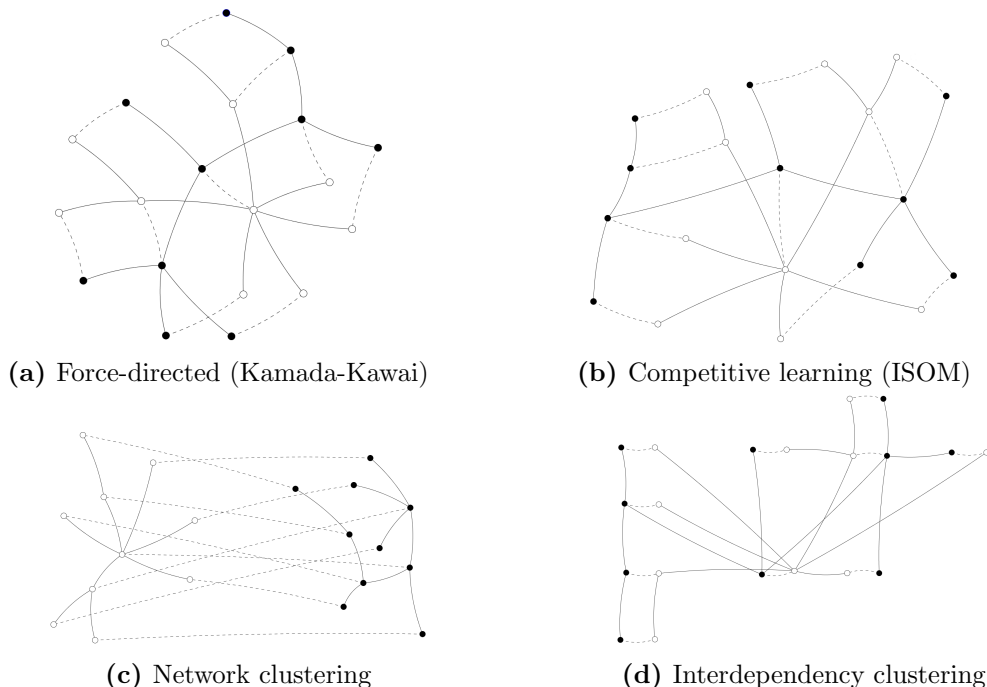


Figure 3.1: An interdependent network, consisting of two scale-free networks connected through interdependencies, with different layout algorithms applied.

Simulation

The definition of robustness as presented in Section 1.1 states that it is defined by the ability of a network to continue performing well when subject to attacks and random failures. Therefore, in order for the tool to visualize the robustness of a network it must have the ability to simulate the cascading effects of these attacks. This simulation must be able to take the current state of the (interdependent) network, make a certain amount of vertices or edges fail, and simulate the networks (and interdependencies) until a steady state is reached.

Failures, as specified by the definition of robustness, are the result of either a random occurrence or a targeted approach. The tool implements three attack strategies to mimic this behavior, but also allows for custom strategies to be implemented.

- Random
- Betweenness
- Manual

The randomized strategy allows for the simulation of random failures, whereas attacks based on the betweenness (Definition 2.9) and manual selection of graph elements (i.e. vertices and edges) represent a targeted approach. Using such a strategy, in combination with the initial state of the network and a certain amount of graph elements to attack, we can compute the network state resulting from these failures.

This failure state is the start of a simulation that applies an iterative process until the (interdependent) network reaches a steady state. The process simulates each network separately using a designated simulator, followed by interdependency resolving between the networks. Such a network simulator takes a network state as input and performs a number of simulation rounds, spreading the failure throughout the network based on its rules. Multiple of these network simulators have been implemented as shown in the following list, an extendable class has also been creating allowing for the addition of custom simulators (including those using the interface to communicate to an external application such as MATLAB).

- Basic, edges adjacent to failed vertices also immediately fail.
- LCC, only the largest connected component remains working, all else fails.
- MATCASC, flow simulation based on the concept defined by Koç et al [25].

The simulation process also needs to synchronize failures between the networks by means of the interdependencies. This interdependency simulation simply transfers failures between connected vertices from different networks, as specified by the IFN model. However, interleaving this step with the network simulations presents two options as shown in Figure 3.2. Either perform all network simulations in parallel followed by a single interdependency synchronization, or sequentially execute the simulation and interdependency synchronization for each individual network.

Both approaches can be useful in simulating cascading failures. Sequential interleaving would work best for networks adhering to scheduled behavior, whereas parallel interleaving is better in cases where all networks are active simultaneously.

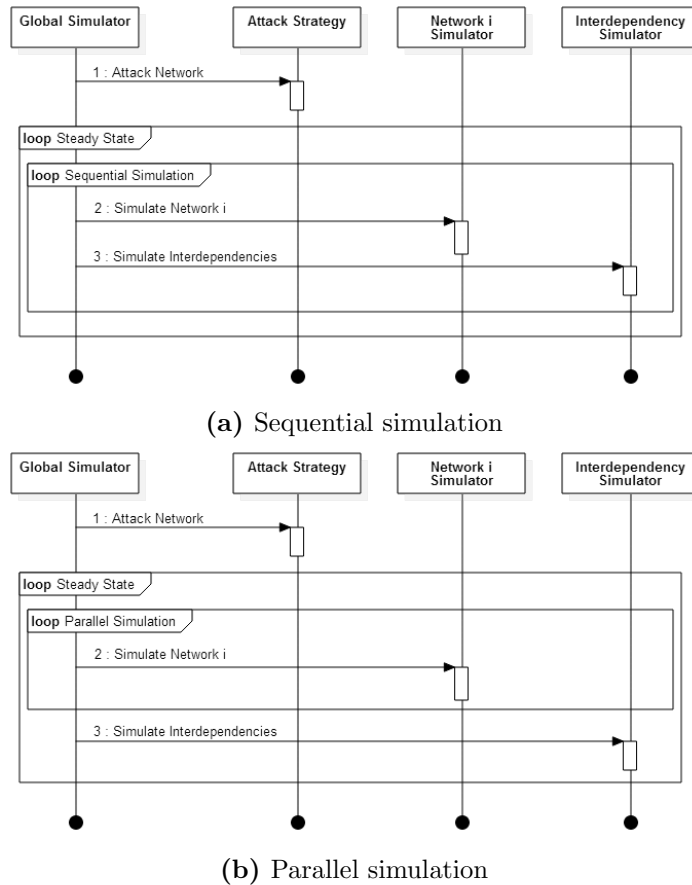


Figure 3.2: Sequence diagrams of simulator interleaving.

Measure analysis

The focus of the tool is network analysis, allowing the user to gather statistics about different aspects of the network. Providing these statistics are the graph measures presented in Chapter 2, which have all been implemented. Although these measures give insight into the structure of the network, they might not give a correct indication of robustness. The relation between these measures and the network robustness will be discussed in Section 3.2.

As for an interdependent network, the graph measures are able to compute a separate value for each of its networks or an overall average. Another option which the tool provides is to consider the interdependent network as a regular network where the interdependencies are represented by edges. Using one of these computation techniques, the tool is able to analyze a graph measure for each stage of the simulation.

After gathering all measure statistics, the data series can be plotted in a chart or statistically analyzed using a variety of techniques such as linear regression and correlation.

Measure improvement

For each of the distance-based measures, as defined in Section 2.1, the tool contains an approximation algorithm aimed at minimizing said measure for a given graph. More details on these algorithms will follow in Chapter 5.

3.2 Measure comparison

In this section we will perform a comparison between the graph measures presented in Chapter 2 to study their effectiveness with respect to capturing the robustness of a given network. The relative robustness of the networks utilized in the comparison is known, allowing for reasoning about the effectiveness of the graph measures. Two sets of networks are considered: a set of small predefined graphs each with a different topology, see Figure 3.3, and a set of large randomly generated graphs. Displayed in Table 3.1 are the graph measure values for all networks in the first set as computed by the tool.

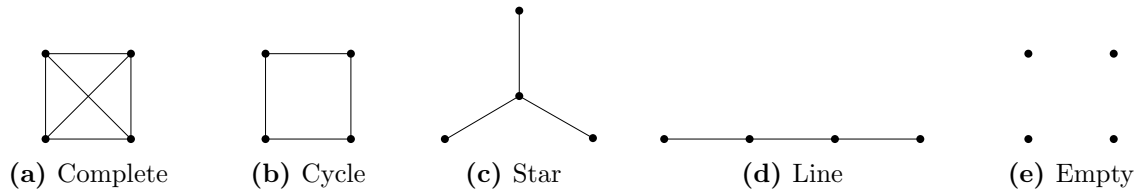


Figure 3.3: Graph topologies consisting of four vertices.

	R	D	D^{avg}	$\bar{\delta}$	\bar{c}	\bar{b}_v	\bar{b}_e	ν	$\bar{\rho}$	λ_2	Ω
Complete	1	1	1	3	1	0	2	4	1	4	3
Cycle	2	2	$1\frac{1}{3}$	2	$\frac{3}{4}$	1	4	4	0	2	5
Star	1	2	$1\frac{1}{2}$	$1\frac{1}{2}$	$\frac{7}{10}$	$1\frac{1}{2}$	6	4	0	1	9
Line	2	3	$1\frac{2}{3}$	$1\frac{1}{2}$	$\frac{5}{8}$	2	$6\frac{2}{3}$	4	0	$\frac{6}{10}$	10
Empty	∞	∞	∞	∞	∞	0	0	1	0	0	∞

Table 3.1: The values of all graph measures for the graphs given in Figure 3.3.

Small topology networks

Intuition says that the graphs in Figure 3.3 are ordered by decreasing robustness. Defined in terms of graph measures, the amount of vertices required to disconnect the network decreases with each graph (i.e. κ_v goes from 3 to 0). Note that $\kappa_v = 1$ for both the star and line graph, but the line graph is less robust because there is more than one way to disconnect the network. Since graph construction is based on the vertex/edge connectivity, this measure has not been added to the table.

The values of all graph measures displayed in Table 3.1 correspond to the intuition, though not all of them can distinguish between graphs. All measures (except radius) confirm that the graphs are indeed in order of decreasing robustness. By looking closely at the classes in the table we can conclude that connectivity measures (i.e. ν and $\bar{\rho}$) are poor at capturing robustness, since there is no significant change in the values between most of the graphs.

All other classes seem to make reasonably good robustness measures. The centrality measures, apart from degree centrality, all have distinct values for each graph making them very suitable. The only downside is the fact that their scaling is of different magnitude and even in opposite directions (degree and closeness versus betweenness). Spectral measures, although proven to capture robustness by Koç et al [27], suffer from the same issues.

The distance-based graph measures also seem to capture robustness reasonably well. The average shortest path length measure is especially good, having distinct values for all graphs (unlike radius and diameter). However, what makes this class as a whole viable as robustness measures is that, contrary to all other classes, the scaling of the measures seems to coincide nicely. This indicates that the measures correlate well together. Hence each measure agrees with the others, and the class presents a combined robustness analysis.

Large randomized networks

Although the analysis on small graphs with different topologies shows some interesting results, it remains to show that the same conclusions still hold when dealing with larger networks based on a more randomized topology. To this end, we will use the random network generation models as discussed in Section 3.1. For each model we will assume three instances, whilst continuously increasing its respective connectivity parameter. This increased connectivity makes a given network relatively more robust than its predecessor, since the additional edges provide more backup connection capabilities.

For each of the network generation models we will assume a vertex count of 100, i.e. $n = 100$, the other parameters are specified according to the following list.

- Erdős-Rényi $G(n, M)$, with M varied between 300, 400, 500.
- Erdős-Rényi $G(n, p)$, with p varied between 0.05, 0.1, 0.15.
- Barabási-Albert, where each new vertex is assigned 1, 2, 3 connections.
- Watts-Strogatz, where the mean degree is varied between 2, 4, 6 (and $\beta = 0.25$).

Displayed in Table 3.2 are the graph measure values for all of these cases as computed by the tool. Each row in the table contains the graph measures averaged over 100 randomly generated networks according to the model.

	R	D	D^{avg}	$\bar{\delta}$	\bar{c}	\bar{b}_v	\bar{b}_e	ν	$\bar{\rho}$	λ_2	Ω
G(n,M)	4	5	2.75	6.00	0.37	173.06	90.69	100	0.06	1.11	2436.41
	3	4	2.43	8.00	0.41	141.40	60.10	100	0.09	1.60	1663.49
	3	4	2.22	10.00	0.45	120.90	43.98	100	0.10	3.24	1213.03
G(n,p)	4	6	2.94	5.12	0.34	192.12	113.72	100	0.05	0.41	3236.62
	3	4	2.24	9.96	0.45	122.82	44.54	100	0.11	1.81	1213.57
	3	3	1.96	14.32	0.51	95.14	27.11	100	0.14	4.52	783.20
BA	6	11	4.74	1.98	0.22	370.56	474.30	100	0.00	0.02	23478.00
	4	6	3.05	3.94	0.33	202.58	153.09	100	0.12	0.63	4705.13
	3	4	2.59	5.88	0.39	157.48	87.24	100	0.17	1.39	2750.27
WS	22	43	15.32	2.00	0.07	1417.50	1516.50	100	0.00	0.00	74686.63
	5	8	4.05	4.00	0.25	302.02	200.51	100	0.28	0.24	4749.21
	4	6	3.04	6.00	0.33	201.60	100.20	100	0.28	0.72	2333.10

Table 3.2: The values of all graph measures for randomly generated graphs.

As can be seen in the table, all measures confirm that the graphs are indeed in order of increasing robustness. The gathered data shows patterns that are similar to those concluded in our previous results. Again, connectivity measures (i.e. ν and $\bar{\rho}$) seem to perform poorly at capturing robustness, since there is no significant change in the values between most of the graphs. All other measures seem to capture robustness reasonably well, providing distinct values albeit scaled differently.

However, the class of distance-based measures again stands out. All three measures analyze a different aspect of shortest path distances on network performance, yet their scaling is still of the same order of magnitude. Hence, we can conclude that distance-based graph measures are viable as robustness measures. This includes using the measures as a basis for network robustness improvement.

Chapter 4

Complexity of improvements

Improving distance-based measures, as defined in Section 2.1, for a given graph can be done by adding previously non-existing edges. However, in many physical networks there is a cost involved when upgrading which discourages the placement of sub-optimal edges. Hence, in an effort to give guarantees on the edge addition, our aim is to minimize the distance-based measures given a limit of k additional edges.

In this chapter we discuss the algorithmic complexity of the minimization problems. We will prove for all of the defined distance-based measures that finding the optimal k additional edges, such that the measure is minimized for a given graph, is NP-complete. In our complexity proofs we will reduce a known NP-complete problem, as defined in Section 4.1, to these minimization problems and show for a given graph that this is the only way to improve the measure optimally.

Thus far, regarding the minimization of distance-based measures, some work has been done analyzing the complexity of both diameter and average shortest path length minimization. The complexity of diameter minimization was initially analyzed by Schoone et al [35], and later refined to a tighter bound by Li et al [28]. Meyerson and Tagiku have also given complexity proofs for several variants of the average shortest path length minimization [32, Theorem 2]. Our proofs for the other measures are based on the technique presented by Schoone et al [35] due to its high adaptability.

In order to differentiate between the measures when working with multiple graphs, which is a common occurrence in the next two chapter, we need to alter our notation slightly. A measure, as defined in Chapter 2, with graph G in subscript notation refers to the value of the measure when applied to G . In particular, the notation of distance-based measures is extended as shown in Table 4.1.

$d_G(u, v)$	weighted shortest path distance between vertices u and v in graph G
$\epsilon_G(v)$	eccentricity of vertex v in graph G
R_G	radius in graph G
D_G	diameter in graph G
$D_G^{avg}(v)$	single source average shortest path length for vertex v in graph G
D_G^{avg}	average shortest path length in graph G

Table 4.1: The extended notation of all distance-based measures.

4.1 Covering by 3-sets

A decision problem is NP-complete when it is both in NP and NP-hard. Hence to prove the NP-completeness for a given problem \mathcal{C} we need to show that

- 1) \mathcal{C} is in NP (a candidate solution to \mathcal{C} is verifiable in polynomial time), and
- 2) Every problem in NP is reducible to \mathcal{C} in polynomial time.

The decision problems of individually minimizing each distance-based measure below a certain bound are proven to be NP-hard through reductions to a specific instance of the set cover problem [24], which is known to be NP-complete.

Initially proven to be NP-complete by Garey and Johnson [18], *exact cover by 3-sets* (x3C) requires that there is no overlap between the selected subsets, all of size three, whose union is equal to the universe. It was even included as one of Richard Karp's 21 NP-complete problems [24].

Definition 4.1. The problem of computing an exact cover by 3-sets:

- NAME: Exact cover by 3-sets (x3C)
 INSTANCE: Given a set X with $|X| = 3q$ and a collection C of 3-element subsets of X .
 PROBLEM: Finding an exact cover of X in C , i.e., a subcollection $C' \subset C$ such that every element of X occurs in exactly one member of C' .

In our reductions we will utilize a variant of the x3C problem, as formulated by Schoone et al [35]. This problem, named *cover by 3-sets* (3C), removes the restriction of the exact cover (i.e. no overlap between the selected subsets) whilst still enforcing subsets of size three. This variation, which is clearly NP-complete due to it being a combination of aspects from both set cover and x3C, is formally defined as follows

Definition 4.2. An equivalent variant of the x3C problem, as presented by Schoone et al [35], which computes a cover by 3-sets:

- NAME: Cover by 3-sets (3C)
 INSTANCE: Given a set X with $|X| = 3q$ and a collection C of 3-element subsets of X such that each element of X occurs in at least one member of C .
 PROBLEM: Finding an exact cover of X in C , i.e. a subcollection $C' \subset C$ with $|C'| = q$, such that every element of X occurs in at least one member of C' .

In order for the 3C problem to be reduced to a graph measure minimization problem, we must represent 3C by means of a graph. This graph, where we will assume that $x_i \in X$ and $c_i \in C$ with $|C| = l$, will at least consist of the following elements

$$V = \{ x_1, \dots, x_{3q}, c_1, \dots, c_l \}$$

$$E = \{ (x_i, c_j) \text{ if } x_i \in C_j \}$$

Note that the 3C problem defines that there should exist a subset $C' \subset C$ containing q vertices from which all vertices in X can be reached. The intuition behind the reductions is to use this property for the correct placement of q shortcut edges, to each vertex in C' , as to reduce the measure defining paths with vertices from X as their endpoints.

In the following sections we prove NP-completeness of the decision problem corresponding to minimizing eccentricity, average shortest path length, radius and diameter, by reduction from 3C. Our reduction for the diameter minimization problem is similar to the reduction by Schoone et al [35]. Nonetheless, the reduction is present in this thesis for the purpose of a coherent and self-contained presentation. Likewise, our complexity proof for the average shortest path length minimization problem expands on the short proofs given by Meyerson and Tagiku [32, Theorem 2].

4.2 Eccentricity

Recall that the eccentricity for a given vertex $v \in V$ is defined as the distance to the furthest vertex in the graph (Definition 2.2). Minimizing such a measure could greatly improve the reachability of v within the graph, which is especially important for targeted improvement of network weaknesses. Formally this minimization problem is defined as follows.

Definition 4.3. The problem of minimizing the eccentricity for a vertex in the given graph:

NAME:	Eccentricity Minimization (EM)
INSTANCE:	An undirected graph $G = (V, E)$, a distance function $d : V \times V \rightarrow \mathbb{R}_0^+$, a vertex $v \in V$, and a bound $k \in \mathbb{N}$.
PROBLEM:	Finding a set F of non-existing edges, with $ F \leq k$, such that eccentricity for vertex v , $\epsilon_{G'}(v)$, is minimized in supergraph $G' = (V, E \cup F)$.

A specific instance of EM is the Unweighted Eccentricity Minimization (U-EM) problem, this unweighted variant has a distance function that only uses unit weights in the computation of its all-pairs shortest paths. All other parameters in this instance remain the same with respect to EM. The corresponding decision problem (in the weighted and unweighted case) is defined as the problem whether the eccentricity $\epsilon_{G'}(v)$ can be reduced to a value below a given threshold $\epsilon' \in \mathbb{N}$. If we are able to prove that the decision problem for U-EM is NP-hard, then the hardness of the other variants naturally follows.

To prove the hardness of the decision problem for U-EM, and hence also EM (through the same reduction, but with the use of a unit weight function), a polynomial time transformation of 3C to the decision problem for U-EM is made. We will construct a graph G with vertex a such that a supergraph G' can be obtained where $\epsilon_{G'}(a) \leq 2$ by adding all edges in set F with $|F| \leq q$, if and only if C contains a subset $C' \subset C$ with $|C'| = q$ such that every element of X occurs in at least one element of C' . Hence we take $v = a$, $\epsilon' = 2$, $k = q$, $|C| = l$, and define $G = (V, E)$ to be as follows (see Figure 4.1).

$$V = \{ x_1, \dots, x_{3k}, c_1, \dots, c_l, b, a \}$$

$$E = \{ (x_i, c_j) \text{ if } x_i \in C_j, (c_i, b) \text{ for } 1 \leq i \leq l, (b, a) \}$$

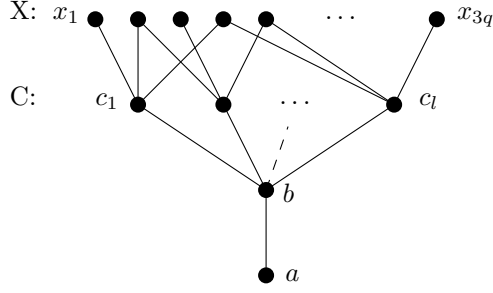


Figure 4.1: Graph $G = (V, E)$ used in the proof of Theorem 4.5.

Using graph G , where $\epsilon_G(a) = 3$, and its supergraph G' , we can relate 3C and U-EM in the following way.

Lemma 4.4. *C contains a subset $C' \subset C$, with $|C'| = q$, such that every element of X occurs in at least one element of C' , if and only if G has a supergraph G' , obtained by adding all q edges from set F , where $\epsilon_{G'}(a) \leq 2$.*

Proof. Suppose C contains such a subset C' . Then let F be the set consisting of all edges (a, c_i) where $c_i \in C'$, it holds that $|F| \leq q$. The resulting supergraph $G' = (V, E \cup F)$ has $\epsilon_{G'}(a) \leq 2$, since the shortest path from a to any $x_j \in X$ is now two due it skipping vertex b .

Conversely, suppose G has such a supergraph $G' = (V, E \cup F)$ with $\epsilon_{G'}(a) \leq 2$. Then for every $x_j \in X$ we must either have an edge (a, x_j) or an edge (a, c_j) for a c_j such that $x_j \in C_j$. Let there be s edges of type (a, x_j) and t edges of type (a, c_j) . Then at most $s + 3t$ x -vertices are at distance two of a . Since we have that $|X| = 3q$ and $s + t \leq q$, i.e. a maximum of q added edges, we can conclude that $s = 0$. This becomes clear in the scenario with $l = q$, where we cannot afford anything else then (a, c_j) edges.

Now, if we define $C' = \{ c_j \in C \mid (a, c_j) \in E' \}$, then we have $|C'| = t \leq q$ and C' is a cover of X . \square

Following from the polynomial time reduction in Lemma 4.4 and the fact that a candidate solution can also be verified in polynomial time, by simply constructing the supergraph and computing the measure, the following theorem follows.

Theorem 4.5. *The decision problems for U-EM and EM are both NP-complete, and thus U-EM and EM are NP-hard.*

4.3 Average Shortest Path

The average shortest path length is a measure that aims at capturing average network performance, instead of best and worst case such as radius and diameter respectively. Hence minimizing such a measure would increase the average distance-based connectivity within the network, which is often wanted in general network improvement. However, due to the averaging this is also a more difficult measure to theoretically analyze.

The measure itself is split into two definitions, a single source variant (Definition 2.5) and a general case (Definition 2.6), just like diameter and radius with respect to eccentricity. Meyerson and Tagiku have given sketches for the complexity proofs of comparable variants, our proofs will expand on these short proofs [32, Theorem 2].

Single source (SSASPM)

The single source variant of average shortest path length minimization is very similar to the EM problem, focusing only on the average reachability of a single vertex within the graph. Hence their definitions are also very similar:

Definition 4.6. The problem of minimizing the average shortest path length for a single source, i.e. vertex, in the given graph:

NAME:	Single Source Average Shortest Path Minimization (SSASPM)
INSTANCE:	An undirected graph $G = (V, E)$, a distance function $d : V \times V \rightarrow \mathbb{R}_0^+$, a vertex $v \in V$, and a bound $k \in \mathbb{N}$.
PROBLEM:	Finding a set F of non-existing edges, with $ F \leq k$, such that average shortest path length for vertex v , $D_{G'}^{avg}(v)$, is minimized in supergraph $G' = (V, E \cup F)$.

As with all minimization problems defined in this chapter, this problem too has an unweighted variant. The unweighted U-SSASPM problem shares all of its parameters with the regular SSASPM except for the distance function, which considers only unit weights in its shortest path computations. Also, we will again consider the corresponding decision problem (in both the weighted and unweighted case), which is defined as the problem whether the measure $D_{G'}^{avg}(v)$ can be reduced to a value below a given threshold $D^{avg'} \in \mathbb{N}$. If we are able to prove that the decision problem for U-SSASPM is NP-hard, then the hardness of the other variants naturally follows.

Proving the hardness of the decision problem for U-SSASPM will also provide us with the same results for SSASPM, using the same reduction with a unit weight function for SSASPM. A polynomial time transformation from 3C to the decision problem for U-SSASPM is made using the same principle as the complexity proof for eccentricity.

Graph G will be constructed with vertex a such that a supergraph G' can be obtained where $D_{G'}^{avg}(a) \leq \frac{1+2l+5q}{n-1}$ with $|C| = l$ by adding all edges in set F with $|F| \leq q$, if and only if C contains a subset $C' \subset C$ with $|C'| = q$ such that every element of X occurs in at least one element of C' . Hence we take $v = a$, $k = q$, $D^{avg'} = \frac{1+2l+5q}{n-1}$, and let $G = (V, E)$ be as defined in the previous section (see Figure 4.1).

Using graph G , where $D_G^{avg}(a) = \frac{1+2l+9q}{n-1}$, and its supergraph G' , we can relate 3C and U-SSASPM in the following way.

Lemma 4.7. C contains a subset $C' \subset C$, with $|C'| = q$, such that every element of X occurs in at least one element of C' , if and only if G has a supergraph G' , obtained by adding all q edges from set F , where $D_{G'}^{avg}(a) \leq \frac{1+2l+5q}{n-1}$.

Proof. Suppose C contains such a subset C' . Then let F be the set consisting of all edges (a, c_i) where $c_i \in C'$, it holds that $|F| \leq q$. The resulting supergraph $G' = (V, E \cup F)$ has $D_{G'}^{avg}(a) \leq \frac{1+2l+5q}{n-1}$, which follows from the following computations with respect to the path lengths starting in vertex a .

$$D_G^{avg}(a) = \frac{1+2l+3 \cdot 3q}{n-1} = \frac{1+2l+9q}{n-1}$$

$$D_{G'}^{avg}(a) = \frac{1+q+2(l-q)+2 \cdot 3q}{n-1} = \frac{1+2l+5q}{n-1}$$

Conversely, suppose G has such a supergraph $G' = (V, E \cup F)$ with $D_{G'}^{avg}(a) \leq \frac{1+2l+5q}{n-1}$. The difference between the measurement value for G and G' is the multiplication with respect to q . Hence the distance between a and each $x_j \in X$, having the only size parameter dependent on q , must be minimized. To this extent, for every $x_j \in X$ we must either have an edge (a, x_j) or an edge (a, c_j) for a c_j such that $x_j \in C_j$.

Let there be s edges of type (a, x_j) and t edges of type (a, c_j) . Then at most $s + 3t$ x -vertices are at a lesser distance to a than before. Since we have that $|X| = 3q$ and $s + t \leq q$, i.e. a maximum of q added edges, we can conclude that $s = 0$. This becomes clear in the scenario with $l = q$, where we cannot afford anything else than (a, c_j) edges.

Now, if we define $C' = \{ c_j \in C \mid (a, c_j) \in E' \}$, then we have $|C'| = t \leq q$ and C' is a cover of X . \square

Since we can verify a candidate solution in polynomial time by computing the measure for a proposed supergraph, and a polynomial time reduction is presented in Lemma 4.4, the following theorem has been proven.

Theorem 4.8. *The decision problems for U-SSASPM and SSASPM are both NP-complete, and thus U-SSASPM and SSASPM are NP-hard.*

General (ASPM)

The general, or complete, variant of average shortest path length minimization is a problem which aims at improving reachability across the entire network since all shortest paths are taken into consideration. Hence, instead of focusing on a single vertex, this definition aims at minimizing a more global measure.

Definition 4.9. The problem of minimizing the average shortest path length the given graph:

NAME:	Average Shortest Path Minimization (ASPM)
INSTANCE:	An undirected graph $G = (V, E)$, a distance function $d : V \times V \rightarrow \mathbb{R}_0^+$, and a bound $k \in \mathbb{N}$.
PROBLEM:	Finding a set F of non-existing edges, with $ F \leq k$, such that average shortest path length $D_{G'}^{avg}$ is minimized for supergraph $G' = (V, E \cup F)$.

A related problem, with a slightly different instance, is the unweighted variant (U-ASPM). In this altered instance, the distance function is replaced with one which only considers unit weights in its shortest path computations. The NP-complete decision problem related to both ASPM and U-ASPM is one with an imposed bound on the maximum measurement value, which is defined as the problem whether the measure $D_{G'}^{avg}$ can be reduced to a value below a given threshold $D^{avg'} \in \mathbb{N}$. Proving the hardness of this decision problem would then also imply the hardness of the related minimization problems.

The hardness of the decision problem for U-ASPM will be proven through a polynomial time transformation from 3C. This reduction will also automatically hold for ASPM, since it can mimic the U-ASPM problem by assuming a unit weight function. Although the reduction is based on the same principle as previous ones, p additional vertices are added to a to balance the tree.

We will construct graph G such that a supergraph G' can be obtained where $D_{G'}^{avg} \leq D_G^{avg} - \frac{8q(p+1)}{n(n-1)}$ by adding all edges in set F with $|F| \leq q$, if and only if C contains a subset $C' \subset C$ with $|C'| = q$ such that every element of X occurs in at least one element of C' .

Hence we take $D^{avg'} = D_G^{avg} - \frac{8q(p+1)}{n(n-1)}$, $k = q$, $|C| = l$, and define $G = (V, E)$ to be as follows (see Figure 4.2).

$$V = \{ x_1, \dots, x_{3q}, c_1, \dots, c_l, \\ b, a, y_1, \dots, y_p \}$$

$$E = \{ (x_i, c_j) \text{ if } x_i \in C_j, (c_i, c_j) \text{ for } 1 \leq i < j \leq l, \\ (c_i, b) \text{ for } 1 \leq i \leq l, (b, a), (a, y_i) \text{ for } 1 \leq i \leq p \}$$

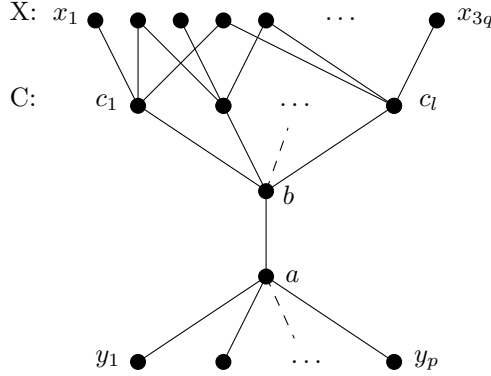


Figure 4.2: Graph $G = (V, E)$ used in the proof of Theorem 4.11.

Using graph G , where we assume a measurement value of D_G^{avg} , and its supergraph G' , we can relate 3C and U-ASPM in the following way.

Lemma 4.10. *C contains a subset $C' \subset C$, with $|C'| = q$, such that every element of X occurs in at least one element of C' , if and only if G has a supergraph G' , obtained by adding all q edges from set F , where $D_{G'}^{avg} \leq D_G^{avg} - \frac{8q(p+1)}{n(n-1)}$.*

Proof. Suppose C contains such a subset C' . Then let F be the set consisting of all edges (a, c_i) where $c_i \in C'$, it holds that $|F| \leq q$. The resulting supergraph $G' = (V, E \cup F)$ has $D_{G'}^{avg} \leq D_G^{avg} - \frac{8q(p+1)}{n(n-1)}$, where the difference of $\frac{8q(p+1)}{n(n-1)}$ can be computed through an addition over all path lengths that are shortened.

- Between X and a , reduced from 3 to 2, for all $2 \cdot 3q$ combinations.
- Between X and y_i , reduced from 4 to 3, for all $2 \cdot 3q \cdot p$ combinations.
- Between C and a , reduced from 2 to 1, for q of the l vertices, with $2 \cdot q$ combinations.
- Between C and y_i , reduced from 3 to 2, for q of the l vertices, with $2 \cdot q \cdot p$ combinations.

Therefore the total reduction in shortest path lengths is:

$$\begin{aligned} & (1 \cdot 2 \cdot 3q) + (1 \cdot 2 \cdot 3q \cdot p) + (1 \cdot 2 \cdot q) + (1 \cdot 2 \cdot q \cdot p) \\ &= 6q + 6q \cdot p + 2q + 2q \cdot p \\ &= 6q(p+1) + 2q(p+1) \\ &= 8q(p+1) \end{aligned}$$

Averaging these results over all combinations leads to the $\frac{8q(p+1)}{n(n-1)}$ difference, hence proving the lemma one way.

Conversely, suppose G has such a supergraph $G' = (V, E \cup F)$ with $D_{G'}^{avg} \leq D_G^{avg} - \frac{8q(p+1)}{n(n-1)}$. Note that at least one edge in F should reduce the sum over all shortest path length by a value of $8(p+1)$. If we assume p to be large enough, say $p > 4(3q+l+1)^2$, then we can conclude that no edges from F are in the subgraph consisting of X , C and b , since $4(3q+l+1)^2$ is an upper bound on the shortest path lengths sum in this subgraph. Even if a single additional edge from F could reduce this to 0, then it would still not come close to an improvement of $8(p+1)$.

Edges in F could also not be between y -vertices, or an y_i and the subgraph consisting of X , C and b , because each edge would only reduce the total average shortest path length by a constant amount, and not quantifiable in p since $p > q$.

Therefore we must have for every $x_j \in X$ either an edge (a, x_j) or an edge (a, c_j) for a c_j such that $x_j \in C_j$. Let there be s edges of type (a, x_j) and t edges of type (a, c_j) . Then at most $s + 3t$ x -vertices are at distance two of a . Since we have that $|X| = 3q$ and $s + t \leq q$, i.e. a maximum of q added edges, we can conclude that $s = 0$. This becomes clear in the scenario with $l = q$, where we cannot afford anything else than (a, c_j) edges.

Now, if we define $C' = \{ c_j \in C \mid (a, c_j) \in E' \}$, then we have $|C'| = t \leq q$ and C' is a cover of X . \square

Following from the fact that we can verify a candidate solution in polynomial time by computing the measure for a proposed supergraph, and a polynomial time reduction is presented in Lemma 4.4, the following theorem has been proven.

Theorem 4.11. *The decision problems for U-ASPM and ASPM are both NP-complete, and thus U-ASPM and ASPM are NP-hard.*

4.4 Radius

The radius distance measure is a function of the eccentricity (Definition 2.3) whose complexity was already discussed in Section 4.2. Due to it being equal to the minimum eccentricity across all vertices in the graph, minimizing such a measure could greatly improve best case performance, making the shorter paths even shorter. This may come into effect when dealing with a centralized vertex, e.g. a server, that spreads information across the network. Moreover, the minimization problem is defined below.

Definition 4.12. The problem of minimizing the radius for a given graph:

NAME:	Radius Minimization (RM)
INSTANCE:	An undirected graph $G = (V, E)$, a distance function $d : V \times V \rightarrow \mathbb{R}_0^+$, and a bound $k \in \mathbb{N}$.
PROBLEM:	Finding a set F of non-existing edges, with $ F \leq k$, such that radius $R_{G'}$ is minimized for supergraph $G' = (V, E \cup F)$.

The RM problem can be instantiated on a graph containing only unit weights on its edges. This specialized variant, also called the Unweighted Radius Minimization (U-RM) problem, has a distance function that only uses unit weights in the computation of its all-pairs shortest paths. All other parameters remain the same with respect to RM. The corresponding decision

problem (in the weighted and unweighted case) is defined as the problem whether the radius $R_{G'}$ can be reduced to a value below a given threshold $R' \in \mathbb{N}$. The hardness of these minimizations will result from the following complexity proof for the decision problem related to U-RM.

A polynomial time transformation from 3C to the decision problem for U-RM is used to prove the hardness, since RM can choose its weight function to mimic the U-RM problem its decision problem must be at least as hard. Graph G will be constructed such that a supergraph G' can be obtained where $R_{G'} \leq 2$ by adding all edges in set F with $|F| \leq q$, if and only if C contains a subset $C' \subset C$ with $|C'| = q$ such that every element of X occurs in at least one element of C' . Hence we take $R' = 2$, $k = q$, $|C| = l$, and define $G = (V, E)$ to be as follows (see Figure 4.3).

$$V = \{ x_1, \dots, x_{3q}, c_1, \dots, c_l, b, a, y_1, \dots, y_{2q+1}, z_1, \dots, z_{2q+1} \}$$

$$E = \{ (x_i, c_j) \text{ if } x_i \in C_j, (c_i, b) \text{ for } 1 \leq i \leq l, (b, a), (a, y_i) \text{ for } 1 \leq i \leq 2q+1, (y_i, z_i) \text{ for } 1 \leq i \leq 2q+1 \}$$

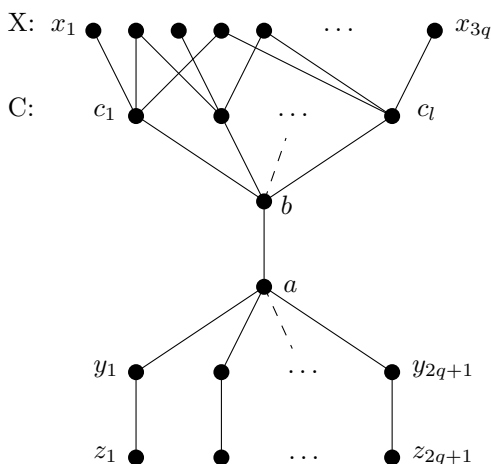


Figure 4.3: Graph $G = (V, E)$ used in the proof of Theorem 4.14.

Using graph G , where $R_G = 3$, and its supergraph G' , we can relate 3C and U-RM in the following way.

Lemma 4.13. *C contains a subset $C' \subset C$, with $|C'| = q$, such that every element of X occurs in at least one element of C' , if and only if G has a supergraph G' , obtained by adding all q edges from set F , where $R_{G'} \leq 2$.*

Proof. Suppose C contains such a subset C' . Then let F be the set consisting of all edges (a, c_i) where $c_i \in C'$, it holds that $|F| \leq q$. The resulting supergraph $G' = (V, E \cup F)$ has $R_{G'} \leq 2$, since the shortest path from a to any other $v \in V$ is now at most two. In graph G the distance from a to any $x_j \in X$ was three, which has been reduced to two in G' due to the edges in F skipping over vertex b .

Conversely, suppose G has such a supergraph $G' = (V, E \cup F)$ with $R_{G'} \leq 2$. Since there are $2q + 1$ y - and z -vertices, there is at least one i for which both y_i and z_i are not incident

with one of the q edges in F . Hence the vertex defining the radius must be a , since it is the only vertex within a distance of two with all z -vertices.

To ensure an eccentricity of two for a , and thus also radius, we must have for every $x_j \in X$ either an edge (a, x_j) or an edge (a, c_j) for a c_j such that $x_j \in C_j$. Let there be s edges of type (a, x_j) and t edges of type (a, c_j) . Then at most $s + 3t$ x -vertices are at distance two of a . Since we have that $|X| = 3q$ and $s + t \leq q$, i.e. a maximum of q added edges, we can conclude that $s = 0$. This becomes clear in the scenario with $l = q$, where we cannot afford anything else than (a, c_j) edges.

Now, if we define $C' = \{ c_j \in C \mid (a, c_j) \in E' \}$, then we have $|C'| = t \leq q$ and C' is a cover of X . \square

The following theorem has now been proven, due to the polynomial time reduction in Lemma 4.13 and the fact that a candidate solution can also be verified in polynomial time by simply constructing the supergraph and computing the measure.

Theorem 4.14. *The decision problems for U-RM and RM are both NP-complete, and thus U-RM and RM are NP-hard.*

4.5 Diameter

The diameter is a distance measure equal to the maximum eccentricity over all vertices in the graph (Definition 2.4). Being a measure that computes the longest path in the graph, it is often used in analyses of worst case network performance. Improving such a measure, hence reducing the longest path, could greatly improve reachability by reducing the distance to outliers. This minimization problem, which tries to decrease the diameter, is defined as follows.

Definition 4.15. The problem of minimizing the radius for a given graph:

NAME:	Diameter Minimization (DM)
INSTANCE:	An undirected graph $G = (V, E)$, a distance function $d : V \times V \rightarrow \mathbb{R}_0^+$, and a bound $k \in \mathbb{N}$.
PROBLEM:	Finding a set F of non-existing edges, with $ F \leq k$, such that diameter $D_{G'}$ is minimized for supergraph $G' = (V, E \cup F)$.

An unweighted instantiation of DM, which is referred to as the Unweighted Diameter Minimization (U-DM) problem, has a distance function that only uses unit weights in the computation of its all-pairs shortest paths. All other parameters in this instance remain the same with respect to DM. The corresponding decision problem (in the weighted and unweighted case) is defined as the problem whether the diameter $D_{G'}$ can be reduced to a value below a given threshold $D' \in \mathbb{N}$. If we are able to prove that the decision problem for U-DM is NP-hard, then the hardness of the other variants naturally follows.

Proving the hardness of this decision problem for U-DM, and thus also DM through the same reduction (with the use of a unit weight function), a polynomial time transformation from 3C to U-DM can be made. This reduction, which inspired all other complexity proofs in this chapter, is similar to the proof as written by Schoone et al [35, Theorem 5.2]. Although some details have been altered to increase familiarity between the proofs in this chapter, substituting our graph for theirs would still result in a completely correct proof.

We will construct a graph G such that a supergraph G' can be obtained where $D_{G'} \leq 3$ by adding all edges in set F with $|F| \leq q$, if and only if C contains a subset $C' \subset C$ with $|C'| = q$ such that every element of X occurs in at least one element of C' . Hence we take $D' = 3$, $k = q$, $|C| = l$, and define $G = (V, E)$ to be as follows (see Figure 4.4).

$$V = \{ x_1, \dots, x_{3q}, c_1, \dots, c_l, \\ b, a, y_1, \dots, y_{2q+1} \}$$

$$E = \{ (x_i, c_j) \text{ if } x_i \in C_j, (c_i, c_j) \text{ for } 1 \leq i < j \leq l, \\ (c_i, b) \text{ for } 1 \leq i \leq l, (b, a), (a, y_i) \text{ for } 1 \leq i \leq 2q+1 \}$$

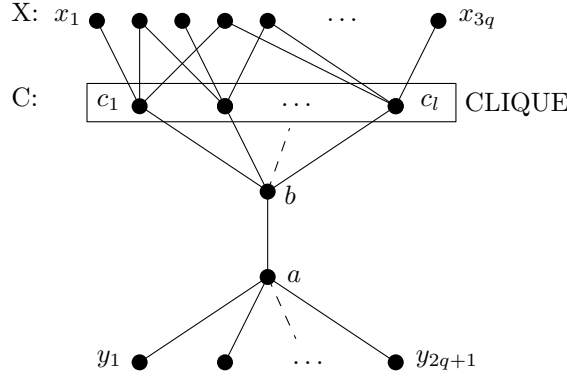


Figure 4.4: Graph $G = (V, E)$ used in the proof of Theorem 4.17.

Using graph G , where $D_G = 4$, and its supergraph G' , we can relate 3C and U-DM in the following way.

Lemma 4.16. *C contains a subset $C' \subset C$, with $|C'| = q$, such that every element of X occurs in at least one element of C' , if and only if G has a supergraph G' , obtained by adding all q edges from set F , where $D_{G'} \leq 3$.*

Proof. Suppose C contains such a subset C' . Then let F be the set consisting of all edges (a, c_i) where $c_i \in C'$, it holds that $|F| \leq q$. The resulting supergraph $G' = (V, E \cup F)$ has $D_{G'} \leq 3$, since the paths that previously defined the diameter, from any x-vertex to any y-vertex, are now shortened to length three. This is due to the addition of edges in F which skip over vertex b . Also note that the clique among all c-vertices is added so the distance between x-vertices does not define the diameter.

Conversely, suppose G has such a supergraph $G' = (V, E \cup F)$ with $D_{G'} \leq 3$. Since there are $2q + 1$ y-vertices, there is at least one y_i which is not incident with one of the q edges in F . Hence the shortest path from y_i to any $x_j \in X$ contains a , and thus the distance from a to any x_j is at most two.

To ensure this distance of at most two from a to any x-vertex we must have for every $x_j \in X$ either an edge (a, x_j) or an edge (a, c_j) for a c_j such that $x_j \in C_j$. Let there be s edges of type (a, x_j) and t edges of type (a, c_j) . Then at most $s + 3t$ x-vertices are at distance two of a . Since we have that $|X| = 3q$ and $s + t \leq q$, i.e. a maximum of q added edges, we can conclude that $s = 0$. This becomes clear in the scenario with $l = q$, where we cannot afford anything else then (a, c_j) edges.

Now, if we define $C' = \{ c_j \in C \mid (a, c_j) \in E' \}$, then we have $|C'| = t \leq q$ and C' is a cover of X . \square

Resulting from the polynomial time reduction presented in Lemma 4.16 and the fact that a candidate solution can also be verified in polynomial time, by simply constructing the supergraph and computing the measure, the following theorem has been proven.

Theorem 4.17. *The decision problems for U-DM and DM are both NP-complete, and thus U-DM and DM are NP-hard.*

Chapter 5

Approximating optimal improvements

As shown in the previous chapter, the problem of minimizing any distance-based measure given a limit on the number of edge additions is NP-hard. Hence no polynomial time algorithm to compute the optimal placement of these edges, i.e. that reduce the given measure the most, exists unless $P = NP$. Therefore, in this chapter, we aim at approximating each of the minimization problems defined in Chapter 4, which includes proving bounds on our approximation ratios with respect to the optimal result.

Since these approximation algorithms are all based on distance measures, it is important that the edge weights of the given graph adhere to the triangle inequality. A definition of the property for graphs is given below.

Definition 5.1. *The triangle inequality* states that, for any three vertices $v_1, v_2, v_3 \in V$ in graph $G = (V, E)$, the distance of the shortest path between v_1 and v_2 is smaller or equal to the sum of distances of the shortest paths from v_1 to v_3 and from v_3 to v_2 :

$$d_G(v_1, v_2) \leq d_G(v_1, v_3) + d_G(v_3, v_2)$$

Note that this inequality always holds with our definition of shortest path distance, because $d_G(v_1, v_3) + d_G(v_3, v_2) < d_G(v_1, v_2)$ contradicts that $d_G(v_1, v_2)$ is the length of the shortest path. Equality only occurs in the case where v_3 is on a shortest path from v_1 to v_2 .

The approximation algorithms presented in this chapter aim at computing a set F of additional edges that reduce the specific distance measure the most. The maximum weight of these new edges is defined by the variable ω , which is specified as follows.

Definition 5.2. The *additional edge weight* ω is given by the maximum edge weight in set F , where F is the set of edges to be added.

If not stated otherwise, all edges in F have weight exactly ω . The definition of the additional edge weight is especially important since it is used in almost all approximation ratio proofs. For simplicity we will also assume a single weight ω for all additional edges. However, one could simply use a preferred value for each separate edge, ω would then represent the maximum value among those weights. In the unweighted case we assume $\omega = 1$.

In the following sections we present for each of the minimization problems defined in Chapter 4 an approximation algorithm as well as a proven bound on the approximated results. Some of these bounds have already been proven elsewhere as is indicated by a citation in the lemma or theorem header. Nonetheless, the proofs are still present in this thesis for the purpose of a coherent and self-contained presentation of the approximation algorithms and bounds. But first, in Section 5.1, we will discuss the clustering algorithm that is a subroutine of all other presented approximation algorithms.

5.1 Clustering

A strategy to minimize shortest path distances in the graph would be to add edges between a single centralized vertex and other vertices centered in more distant groups. Using this technique the connectivity will be improved, since all vertices are then able to better reach the centralized vertex and continue from there. An important factor in this approach is the distance from a vertex to the center of the group it belongs to, which has to be minimized to ensure the shortest possible path to the centralized vertex.

Identifying these groups, or clusters rather, can be done using a clustering algorithm, particularly one solving the *k-center problem*. The k-center or facility location problem is a combinatorial optimization problem that wants to partition the set of vertices into k subsets, each with a center, such that for all vertices in the graph the maximum distance to the closest center is minimized. Formally it is defined as follows.

Definition 5.3. The *k-center* or *facility location* problem, given an undirected graph $G = (V, E)$ with distances satisfying the triangle inequality, is to find a subset $S \subseteq V$ with $|S| = k$ while minimizing the objective function:

$$\max_{v \in V} \min_{s \in S} d_G(v, s)$$

The k-center problem is known to be NP-complete. In fact, it has been proven that the problem cannot be approximated with a factor better than 2, unless $P = NP$ [40]. However, a 2-approximation algorithm was given by Gonzalez [19], which will be discussed in more detail in this section.

The algorithm presented by Gonzalez takes as input a graph $G = (V, E)$, a distance function $d_G : V \times V \rightarrow \mathbb{R}_0^+$, and a value $k \in \mathbb{N}^+$. The algorithm returns a k-clustering $\langle B_1, \dots, B_k \rangle$ with heads, i.e. centers, h_i for each $1 \leq i \leq k$. First h_1 is chosen arbitrarily, or given as a parameter, and $B_1 = V$. Then, for every $l = 2, \dots, k$, a vertex that maximizes the distance to its clusters head is selected to be head h_l , and all other vertices are redistributed among the clusters by assigning them to those whose head is closest. We give pseudocode for this algorithm below (Algorithm 1).

Note that Algorithm 1 (i.e. CLUSTER) only returns the heads of the resulting k-clustering, since only these vertices are used in the distance measure minimization. Also, given that the all-pairs shortest path distance function is pre-computed and given as a parameter, the time complexity of Algorithm CLUSTER is given by the following theorem.

Algorithm 1: Gonzalez approximation algorithm for the k-center problem [19].

Input: $G = (V, E)$, the graph that needs to be clustered

$d_G : V \times V \rightarrow \mathbb{R}_0^+$, the all-pairs distance function of graph G

$v \in V$, the initial cluster head

$k \in \mathbb{N}^+$, the required number of cluster

Output: H , an array of cluster heads approximation the optimal k-split

CLUSTER(G, d, v, k)

- (1) $h_1 \leftarrow v$
- (2) $B_1 \leftarrow \{v_1, \dots, v_n\}$
- (3) **for** $l = 2$ **to** k
- (4) $x \leftarrow \max \{ d_G(v_i, h_j) \mid v_i \in B_j \wedge 1 \leq j < l \}$
- (5) $Y \leftarrow \{ v_i \mid d_G(v_i, h_j) = x \wedge v_i \in B_j \wedge 1 \leq j < l \}$
- (6) Let u be a vertex from Y
- (7) $h_l \leftarrow u$
- (8) $B_l \leftarrow \{u\}$
- (9) **foreach** $v_i \in (B_1 \cup \dots \cup B_{l-1})$
- (10) Let j be such that $v_i \in B_j$
- (11) **if** $d_G(v_i, h_l) \leq d_G(v_i, h_j)$
- (12) $B_j \leftarrow B_j \setminus \{v_i\}$
- (13) $B_l \leftarrow B_l \cup \{v_i\}$
- (14) **return** (h_1, \dots, h_k)

Theorem 5.4 ([19, Theorem 2.3]). *The time complexity of algorithm CLUSTER is $O(kn)$.*

Proof. While lines 1 and 2 take constant time, the loop from lines 3 to 13 has $O(k)$ iterations.. The pseudocode in this loop contains multiple lines with a $O(n)$ time complexity, i.e. line 4, 5 and the loop starting on line 9. Hence the total time complexity is indeed $O(kn)$. \square

Using a $(k+1)$ -clique, Gonzalez proved the following approximation ratio.

Theorem 5.5 ([19, Theorem 2.2]). *Algorithm CLUSTER computes a 2-approximation of the objective function.*

5.2 Eccentricity

In Section 4.2 we proved that minimizing the eccentricity is NP-hard. This does not only hold for the general version EM of this problem but also for its unit weight variant U-EM. This section will show how to approximate the general problem within a ratio of $2 + \frac{\omega}{\epsilon^*(v)}$, which for the unit weight case implies a bound of $2 + \frac{1}{\epsilon^*(v)}$, when given an optimal resulting eccentricity of $\epsilon^*(v)$.

Our approach to approximating the EM and U-EM problem, which was already shortly discussed in the previous section, is to connect v to other vertices which are central in a distant cluster. The intuition is that this technique would significantly decrease the path length between v and all vertices in these clusters. Although this approach might not yield an optimal result, it can be proven by the following lemma that adding edges incident to v is optimal. A similar lemma has also been discussed by Meyerson and Tagiku [32, Lemma 1].

Lemma 5.6. *When improving the eccentricity of vertex $v \in V$, there exists an optimal set F^* such that each edge $e \in F^*$ is incident on v .*

Proof. Given an optimal set of additional edges F^* , with $(s, t) \in F^*$. Let π_1 be a shortest path from v to x that traverses (s, t) in the st direction, and π_2 be a shortest path from v to y that traverses (s, t) in the ts direction. Then path π_3 that starts in v , follows π_1 until s and afterwards follows π_2 , never crosses (s, t) . Path π_3 can also never be longer than π_2 , otherwise there would exist a path from v to x that is shorter than π_1 . This case, illustrated by Figure 5.1, would contradict the assumption that π_1 is the shortest path from v to x . Thus showing that (s, t) has an implicit orientation such that it is only ever used in the correct direction.

The contradiction becomes clear when assuming $w(s, t) = 0$ in the following deduction.

$$\begin{aligned} \pi_3 &> \pi_2 \\ d(v, s) + d(s, y) &> d(v, t) + w(t, s) + d(s, y) \\ d(v, s) + d(s, y) &> d(v, t) + d(s, y) \\ d(v, s) &> d(v, t) \\ d(v, s) + w(s, t) &> d(v, t) \\ d(v, s) + w(s, t) + d(t, x) &> d(v, t) + d(t, x) \\ \pi_1 &> d(v, t) + d(t, x) \end{aligned}$$

Since edge (s, t) is only used in one direction, say from s to t , then moving s closer to v would only increase the improvement in eccentricity. Hence the solution $(F^* \setminus \{(s, t)\}) \cup \{(v, t)\}$ is at least as good. Doing this for all other edges would ensure that F^* only contains edges incident on v , proving the lemma. \square

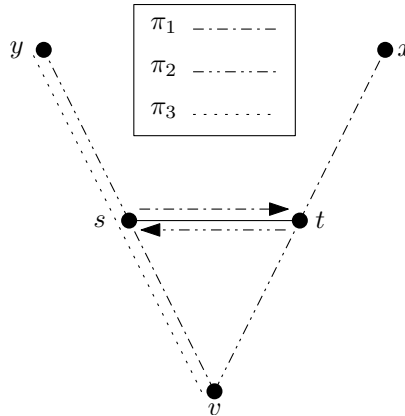


Figure 5.1: Graph illustrating the paths used in the proof of Lemma 5.6

Lemma 5.6 indicates that it would be optimal to have all k edges in F incident to v , the vertex whose eccentricity we are aiming to improve. Therefore, our approach is to partition the graph into $k + 1$ clusters, vertex v being the head of the initial cluster, and defining F to be the set of k edges from v to all other cluster heads. This technique ensures that the added edges connect v to k well separated vertices in the graph, hence improving the eccentricity. Formally, this approach is described by the following algorithm.

Algorithm 2: Approximation algorithm for single source distance measure improvement.

Input: $G = (V, E)$, graph with vertex whose measure needs improvement

$d_G : V \times V \rightarrow \mathbb{R}_0^+$, the all-pairs distance function of graph G

$v \in V$, the vertex whose measure needs improvement

$k \in \mathbb{N}^+$, a bound on the number of added edges

Output: F , a set of non-existing edges to improve the measure over v

APPROXSOURCE(G, d, v, k)

(1) $H \leftarrow \text{CLUSTER}(G, d, v, k + 1)$

(2) $F \leftarrow \{ (h_1, h_i) \mid 1 < i \leq k + 1 \}$

(3) **return** F

Theorem 5.7. *The time complexity of algorithm APPROXSOURCE is $O(kn)$.*

Proof. The first line of the APPROXSOURCE algorithm performs a call to the CLUSTER algorithm, hence taking $O((k + 1)n) = O(kn + n) = O(kn)$ time, as proven in Theorem 5.4. The second line uses these results to build a set of size k , taking $O(k)$ time to do so. Therefore, in total, the algorithm has a running time of $O(kn)$. \square

Before we can prove the approximation ratio of Algorithm APPROXSOURCE, or any other algorithms in this chapter, we must first define some basic notation. Specifically, for a graph $G = (V, E)$, we denote the radius of $U \subseteq V$ in G by $R_G(U) = \min_{v \in V} \max_{u \in U} d_G(v, u)$. Note that for a cluster B_i it holds that $\max_{v \in B_i} d_G(h_1, v) = R_G(B_i)$, in other words the head is the most central vertex in the cluster.

Also, a set of vertices from G , $\{v_1, \dots, v_\ell\}$, is an *independent set* of G if and only if $(v_i, v_j) \notin E$ for all $1 \leq i, j \leq \ell$. Such an independent set is maximum if for every other independent set $\{v_1, \dots, v_{\ell'}\}$ of G we have that $\ell \geq \ell'$. Let $\alpha(G)$ denote the cardinality of the maximum independent set of G . And finally, for $\tau \in \mathbb{R}_0^+$, let $E_\tau = \{(u, v) \mid u, v \in V \wedge d_G(u, v) \leq \tau\}$ and $G_\tau = (V, E_\tau)$. The intuition is that $\alpha(G_\tau)$ is the maximum size of the set of vertices such that all pairs have a distance larger than τ .

With this new notation we can introduce the following two lemmata.

Lemma 5.8 ([7, Lemma 1]). *Let $G = (V, E)$ be a graph, and $\tau \in \mathbb{R}_0^+$. If $\alpha(G_\tau) \leq k$, then the CLUSTER algorithm on input G and k computes a k -clustering of V such that $R_G(B_i) \leq \tau$ for all $1 \leq i \leq k$.*

Proof. First, we will show by contradiction that when given a graph $G = (V, E)$ with $\alpha(G_\tau) \leq k$, all vertices are within distance τ of their cluster head $\max_{v \in V} \min_{1 \leq i \leq k} d_G(h_i, v) \leq \tau$. Hence, for the sake contradiction, assume that $\gamma = \max_{v \in V} \min_{1 \leq i \leq k} d_G(h_i, v) > \tau$. Let $u \in V$ be any vertex such that $u \in \{v \mid v \in V \wedge \min_{1 \leq i \leq k} d_G(h_i, v) = \gamma\}$.

By construction of the clustering, $d_G(h_i, h_j) > \tau$ for every $1 \leq i, j \leq k$ with $i \neq j$. Therefore, we then have that $\{h_1, \dots, h_k\} \cup \{u\}$ is an independent set of $\alpha(G_\tau)$ with cardinality $k + 1$, thus contradicting the assumption that $\alpha(G_\tau) \leq k$. From this we can conclude the following for every $1 \leq i \leq k$

$$R_G(B_i) \leq \max_{v \in B_i} d_G(h_i, v) \leq \max_{v \in V} \min_{1 \leq j \leq k} d_G(h_j, v) \leq \tau.$$

\square

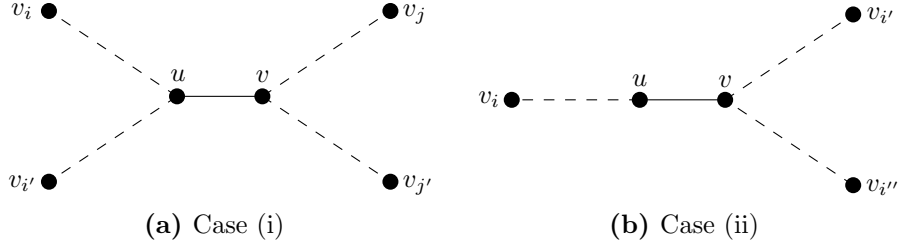


Figure 5.2: The two cases of the proof of Lemma 5.9.

Lemma 5.9 ([7, Lemma 2]). *Let $G = (V, E)$ be a graph, and $\tau \in \mathbb{R}_0^+$. For every $e \notin E$, with $G' = (V, E \cup \{e\})$, $\alpha(G'_\tau) \geq \alpha(G_\tau) - 1$.*

Proof. Let $\{v_1, \dots, v_\ell\}$ be an independent set in G_τ . Then, for the sake of contradiction, assume an edge $(u, v) \notin E$ exists, with $G' = (V, E \cup \{(u, v)\})$, such that $\alpha(G'_\tau) < \ell - 1$. This implies that, in the independent set, (i) there exist four distinct indices $1 \leq i, i', j, j' \leq \ell$ such that $(v_i, v_j), (v_{i'}, v_{j'}) \in E_\tau$, or (ii) there exist three distinct indices $1 \leq i, i', i'' \leq \ell$ such that $(v_i, v_{i'}), (v_{i'}, v_{i''}), (v_{i''}, v_i) \in E_\tau$. Both cases are displayed in Figure 5.2.

In the first case, since $d_G(v_i, v_j), d_G(v_{i'}, v_{j'}) > \tau$ and $d_{G'}(v_i, v_j), d_{G'}(v_{i'}, v_{j'}) \leq \tau$, we can conclude that in G' both the shortest path from v_i to v_j and the shortest path from $v_{i'}$ to $v_{j'}$ must pass through edge (u, v) . We also have that $d_G(v_i, v_{i'}), d_G(v_j, v_{j'}) > \tau$, since all four vertices are part of the independent set. Therefore, without of loss generality, we have the following

$$\begin{aligned}
 d_G(v_i, u) + w(u, v) + d_G(v, v_j) &\leq \tau \\
 d_G(v_{i'}, u) + w(u, v) + d_G(v, v_{j'}) &\leq \tau \\
 \tau &< d_G(v_i, u) + d_G(u, v_{i'}) \\
 \tau &< d_G(v_j, v) + d_G(v, v_{j'}).
 \end{aligned}$$

The sum of these inequalities results in $2 \cdot w(u, v) < 0$, a contradiction.

In the second case, since $d_G(v_i, v_{i'}), d_G(v_{i'}, v_{i''}), d_G(v_{i''}, v_i) > \tau$ and $d_{G'}(v_i, v_{i'}), d_{G'}(v_{i'}, v_{i''}), d_{G'}(v_{i''}, v_i) \leq \tau$, we can conclude that any shortest path in G' between any pair of the three vertices $v_i, v_{i'}, v_{i''}$ has to go through edge (u, v) . Hence, two of the three endpoints must be closer to either u or v than the other one. Without loss of generality, assume that $d_G(v_{i'}, v) \leq d_G(v_{i'}, u)$ and $d_G(v_{i''}, v) \leq d_G(v_{i''}, u)$. Also, we have that $d_G(v_i, v_{i'}), d_G(v_j, v_{j'}) > \tau$, since all four vertices are part of the independent set. As a consequence we can deduce the following

$$\begin{aligned}
 d_G(v_{i'}, v_{i''}) &\leq d_G(v_{i'}, v) + d_G(v_{i''}, v) \\
 &< \min\{d_G(v_{i'}, v) + w(v, u) \\
 &= d_G(u, v_{i''}), d_G(v_{i'}, u) + w(u, v) = d_G(v, v_{i''})\} \\
 &= d_{G'}(v_{i'}, v_{i''}) \\
 &\leq \tau.
 \end{aligned}$$

Therefore $(v_{i'}, v_{i''}) \in E_\tau$, which is a contradiction. □

To prove the approximation ratio of Algorithm APPROXSOURCE we need to introduce two additional lemmata with respect to the lower bound of the measures.

Lemma 5.10. *For a given graph $G = (V, E)$ it holds that $D_G \leq 2 \cdot R_G$.*

Proof. For the sake of contradiction, assume $2 \cdot R_G < D_G$, and let $v \in V$ be the vertex such that $\epsilon_G(v) = R_G$ (i.e. the vertex defining the radius). Therefore, by definition of the eccentricity, $d_G(v, u) \leq R_G$ for all $u \in V$.

Let vertices $s, t \in V$ be the endpoints defining the diameter; $d_G(s, t) = D_G$. Then the path from s to t passing through v is of length $d_G(s, v) + d_G(v, t) \leq 2 \cdot R_G$. Since we assumed $2 \cdot R_G < D_G$, this contradicts the assumption that the current diameter is defined by a shortest path. Thus the lemma is proven. \square

Lemma 5.11. *For a given graph $G = (V, E)$, with $v \in V$, it holds that $D_G \leq 2 \cdot \epsilon_G(v)$.*

Proof. Since the radius is defined as the minimum eccentricity we know that $R_G \leq \epsilon_G(v)$. From this and Lemma 5.10 we can conclude

$$\begin{aligned} R_G &\leq \epsilon_G(v) \\ 2 \cdot R_G &\leq 2 \cdot \epsilon_G(v) \\ D_G &\leq 2 \cdot \epsilon_G(v). \end{aligned}$$

\square

Theorem 5.12. *Algorithm APPROXSOURCE computes a $\left(2 + \frac{\omega}{\epsilon^*(v)}\right)$ -approximation for the U-EM and EM problem.*

Proof. Let F^* be an optimal solution for EM in the instance $G = (V, E)$, $v \in V$, and $k \in \mathbb{N}$, hence $|F^*| \leq k$. For graph $G^* = (V, E \cup F^*)$ we denote $D^* = D_{G^*}$ and $\epsilon^*(v) = \epsilon_{G^*}(v)$. Also, by definition $\alpha(G_{D^*}^*) = 1$, since each vertex in G^* can reach all other vertices within a diameters distance. Using Lemma 5.9 repeatedly will then result in $\alpha(G_{D^*}^*) \geq \alpha(G_{D^*}^*) - |F^*|$, which can be rewritten as $\alpha(G_{D^*}^*) \leq k + 1$. As a consequence of Lemma 5.8, the CLUSTER algorithm computes a $(k + 1)$ -clustering $\langle B_1, \dots, B_{k+1} \rangle$ such that $R_G(B_i) \leq D^*$ for all $1 \leq i \leq k + 1$. With the use of Lemma 5.11 this becomes $R_G(B_i) \leq D^* \leq 2 \cdot \epsilon^*(v)$.

Now consider set $F = \{(h_1, h_i) \mid 2 \leq i \leq k + 1\}$, with $|F| \leq k$, as computed in algorithm APPROXSOURCE. We will denote the resulting graph as $G' = (V, E \cup F)$. Every vertex in G' is at distance at most $2 \cdot \epsilon^*(v) + \omega$ from v , as is illustrated in Figure 5.3. Therefore, $\epsilon_{G'}(v) \leq 2 \cdot \epsilon^*(v) + \omega$, and we have an approximation ratio of $2 + \frac{\omega}{\epsilon^*(v)}$.

This approximation ratio also holds for U-EM by the same analysis. Moreover, due to the fact that $\omega = 1$ in this case, the approximation ratio is improved to $2 + \frac{1}{\epsilon^*(v)}$. \square

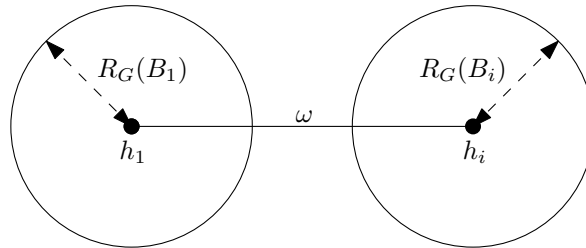


Figure 5.3: Graph illustrating the eccentricity bound proven in Theorem 5.12.

5.3 Average Shortest Path

In Section 4.3 we proved that minimizing the (single source) average shortest path length is NP-hard. This does not only hold for the general version of this problem but also for its unit weight variants U-ASPM and U-SSASPM. This section will show how to approximate the single source case within a ratio of 2 and for the general version a ratio of 4. However, instead of minimizing $D^{avg}(v)$ and D^{avg} , as presented in Definition 2.5 and 2.6 respectively, all proofs will be based on the measures without the added division over all vertex combination.

$$D^{avg}(v) : \sum_{u \in V} d(u, v)$$

$$D^{avg} : \sum_{u \in V} \sum_{v \in V} d(u, v)$$

Minimizing these distance functions will give the same results as minimizing the original definitions of $D^{avg}(v)$ and D^{avg} , since the omitted division does not change in value. First consider the single source variant.

Single source (SSASPM)

Our approach to approximating the SSASPM problem is similar to the one used in minimizing the eccentricity. As discussed by Meyerson and Tagiku in their analysis of the SSASPM problem, and Lemma 5.6 in the previous section, adding edges incident to the source vertex is optimal [32, Lemma 1]. This holds for both the eccentricity and the single source average shortest path length, since both rely on shortest paths with the source as an endpoint.

Therefore, adding edges from source vertex v to other vertices which are central in distant clusters is a technique that still applies. With these connections to all other vertices, through the cluster heads, the average lengths of shortest path starting in v should be decreased significantly. Hence, we will apply Algorithm 2 (i.e. APPROXSOURCE) to the SSASPM problem and prove a different approximation bound with the following theorem.

Theorem 5.13. *Algorithm APPROXSOURCE computes a 2-approximation for the SSASPM problem.*

Proof. Let F^* be an optimal solution for SSASPM in the instance $G = (V, E)$, $v \in V$, and $k \in \mathbb{N}$, hence $|F^*| \leq k$. By Lemma 5.6 we know that all edges contained in F^* are incident to v . Therefore, we only need to find k endpoints for the edges such that the distance for each vertex $u \in V$ to its closest endpoint is minimized, since this also minimizes the average shortest path from v to all u .

The APPROXSOURCE algorithm uses k-center clustering algorithm CLUSTER to compute the endpoints, afterwards the edges from v to these endpoints are stored in set F . The cluster heads, i.e. endpoints, returned by this algorithm are aimed at minimizing the maximum distance to the closest head for all $u \in V$, as specified by the objective function in Definition 5.3. Since the average distance to such an endpoint is always smaller or equal than the maximum distance to endpoints overall, we can assume the same 2-approximation ratio as proven for the clustering algorithm in Theorem 5.5.

Note that distance ω of all added edges is included in both set F and F^* , therefore it is not a part of the approximation ratio. Hence, due to the chosen clustering algorithm, algorithm APPROXSOURCE computes set F which provides a 2-approximation of F^* . Both in the weighted SSASPM and unweighted U-SSASPM problem. \square

General (ASPM)

Thus far we have only dealt with minimizing single source measures, where all additional edges are incident to the source itself. In a more general graph measure such as the average shortest path length, all combinations of vertices must be taken into consideration. To this end, additional edges could be placed all over the graph, not all being incident to the same vertex.

However, an intuitive idea would be to apply the single source algorithm to all vertices in the graph and store the best solution thus far. To do this we must first prove the following lemma, which claims that there exists a vertex in the graph that is sufficiently close to all other vertices.

Lemma 5.14 ([32, Lemma 2]). *For a given graph $G = (V, E)$, there exists a source vertex $v \in V$ for which algorithm APPROXSOURCE returns a 4-approximation for the ASPM problem.*

Proof. Let F^* be an optimal solution for ASPM in the instance $G = (V, E)$ and $k \in \mathbb{N}$, hence $|F^*| \leq k$. We denote $G^* = (V, E \cup F^*)$ to be the graph with all optimal edges added. For this graph, the average value of $D_{G^*}^{avg}(u)$ over all $u \in V$ is $\frac{1}{n}D_{G^*}^{avg}$, hence at least one vertex v must not exceed this average.

Therefore, we will compute a set F , with $G' = (V, E \cup F)$, aimed at minimizing $D_{G'}^{avg}(v)$. Algorithm APPROXSOURCE allows us to do so within an approximation ratio of a , as proven by Theorem 5.13. Using this approximation ratio we can deduce

$$D_{G'}^{avg}(v) \leq 2 \cdot D_{G^*}^{avg}(v) \leq 2 \cdot \frac{1}{n}D_{G^*}^{avg}. \quad (5.1)$$

A bound $D_{G'}^{avg}$ in terms of $D_{G'}^{avg}(v)$ can be obtained by applying the triangle inequality given in Definition 5.1, which specifies for vertices $x, y, z \in V$ that $d(x, z) \leq d(x, y) + d(y, z)$. When summing this inequality over all vertex pairs, the distances between intermediate vertex y and all other vertices in V occurs twice on the right hand side of the inequalities (once as x and once as z) for all n possible choices of y . Hence, the bound defined as

$$D_{G'}^{avg} \leq 2n \cdot D_{G'}^{avg}(v). \quad (5.2)$$

These two equations can be combined to finalize our proof.

$$\begin{aligned} D_{G'}^{avg} &\leq 2n \cdot D_{G'}^{avg}(v) \\ &\leq 4 \cdot D_{G^*}^{avg} \end{aligned}$$

□

Therefore, given a source vertex v with $D_{G^*}^{avg}(v) \leq \frac{1}{n}D_{G^*}^{avg}$ as stated in Lemma 5.14, we are able to treat the general ASPM problem as a single source problem (SSASPM) and apply algorithm APPROXSOURCE to produce a 4-approximation. However, finding this vertex v requires knowledge of F^* , which we do not have. Hence, we must try all vertices and take the best solution. This approach is captured by the following algorithm.

Algorithm 3: Approximation algorithm for the ASPM problem.

Input: $G = (V, E)$, graph with vertex whose ASP length needs improvement

$d_G : V \times V \rightarrow \mathbb{R}_0^+$, the all-pairs distance function of graph G

$k \in \mathbb{N}^+$, a bound on the number of added edges

Output: F , a set of non-existing edges to improve the ASP length.

APPROXASP(G, d, k, M)

- (1) $F \leftarrow \emptyset$
- (2) $m \leftarrow \infty$
- (3) **foreach** $v \in V$
- (4) $F' \leftarrow \text{APPROXSOURCE}(G, d, v, k)$
- (5) $G' \leftarrow (V, E \cup F')$
- (6) **if** $D_{G'}^{avg} < m$
- (7) $F \leftarrow F'$
- (8) $m \leftarrow D_{G'}^{avg}$
- (9) **return** F

The time complexity of Algorithm 3 (i.e. APPROXASP), although polynomial, is strongly dependent on the method used to compute $D_{G'}^{avg}$ as shown in the theorem below.

Theorem 5.15. *The time complexity of algorithm APPROXASP is $O(n^4)$.*

Proof. The APPROXASP algorithm mainly consists of a loop over all n vertices in V . For each of these vertices the single source measure minimization is approximated using algorithm APPROXSOURCE, which has a running time of $O(nk)$ as proven by Theorem 5.7. Moreover, all k edges returned by the measure minimization are added to a new graph, taking $O(k)$ time. Afterwards the average shortest path length measure of the resulting graph is computed, which is done in $O(n^3)$ time by the Floyd-Warshall all-pairs shortest path algorithm [15]. However, one could also apply Johnson's algorithm when dealing with sparse graphs, having a running time of $O(n^2 \log n + nm)$ [21]. In our analysis we will assume the Floyd-Warshall algorithm is used, due to its general applicability. Finally, a comparison is made with the best solution thus far, only taking constant time.

Therefore, within the main loop, the Floyd-Warshall algorithm has the highest time complexity of all actions. When multiplying this by the number of iterations we get a total running time of $O(n \cdot n^3) = O(n^4)$. \square

Theorem 5.16. *Algorithm APPROXASP computes a 4-approximation for the ASPM problem.*

Proof. Lemma 5.14 states that, for a given graph $G = (V, E)$, a vertex $v \in V$ exists such that algorithm APPROXSOURCE on v returns a 4-approximation for the ASPM problem. Since the APPROXASP algorithm applies APPROXSOURCE to all vertices in V , storing the best solution seen thus far, the result must be at least a 4-approximation. This also holds for the unweighted U-ASPM variant. \square

5.4 Radius

Minimizing the radius is another optimization problem based on a measure defined on the overall graph. The definition and complexity proof of the RM problem, and its unweighted counterpart U-RM, can be found in Section 4.4. An algorithm approximating these optimization problems with ratio $2 + \frac{\omega}{R^*}$, for an optimal resulting radius of R^* , will be presented below.

As with general average shortest path length minimization, our intuition will be to minimize the measure for only a single source vertex. However, due to the definition of the radius, we can prove a much tighter bound on the approximation ratio. The following lemma will give an indication of this bound with respect to the single source algorithm.

Lemma 5.17. *For a given graph $G = (V, E)$, assuming that algorithm APPROXSOURCE returns an α -approximation for the EM problem, there exists a source vertex $v \in V$ for which algorithm APPROXSOURCE returns an α -approximation for the RM problem.*

Proof. Let F^* be an optimal solution for RM in the instance $G = (V, E)$ and $k \in \mathbb{N}$, hence $|F^*| \leq k$. For graph $G^* = (V, E \cup F^*)$ we denote $R^* = R_{G^*}$. Then by definition of the radius there must exist a vertex $v \in V$, where $\epsilon^*(v) = \epsilon_{G^*}(v)$, such that $\epsilon^*(v) = R^*$. Assuming that algorithm APPROXSOURCE returns an α -approximation for $\epsilon^*(v)$, in the form of an edge set F with $G' = (V, E \cup F)$ and $\epsilon_{G'}(v)$, we can conclude that APPROXSOURCE also returns an α -approximation for R^* and therefore the RM problem on the same instance. \square

As a result of Lemma 5.17, we are able to treat RM as a single source problem given a source vertex v . However, finding this vertex v requires knowledge of F^* , which we do not have. Hence, we must iterate over all vertices in the graph and store the best solution we come across.

Algorithm 4: Approximation algorithm for the RM problem.

Input: $G = (V, E)$, graph with vertex whose radius needs improvement

$d_G : V \times V \rightarrow \mathbb{R}_0^+$, the all-pairs distance function of graph G

$k \in \mathbb{N}^+$, a bound on the number of added edges

Output: F , a set of non-existing edges to improve the radius.

APPROXRAD(G, d, k, M)

- (1) $F \leftarrow \emptyset$
- (2) $m \leftarrow \infty$
- (3) **foreach** $v \in V$
- (4) $F' \leftarrow \text{APPROXSOURCE}(G, d, v, k)$
- (5) $G' \leftarrow (V, E \cup F')$
- (6) **if** $R_{G'} < m$
- (7) $F \leftarrow F'$
- (8) $m \leftarrow R_{G'}$
- (9) **return** F

Similar to the average shortest path length minimization algorithm, the time complexity of Algorithm 4 (i.e. APPROXRAD) strongly depends on the methods used to compute the radius. This is illustrated by the theorem proven below.

Theorem 5.18. *The time complexity of algorithm APPROXRAD is $O(n^4)$.*

Proof. Note that the APPROXRAD algorithm is very similar to APPROXASP, the main difference being the computation of measure $R_{G'}$. This is by no means a coincidence, more details regarding a generalization will be discussed in Section 5.5. Computing the radius measure for graph G' can also be done using the same all-pairs shortest path algorithms used to compute the average shortest path length, either the Floyd-Warshall algorithm [15] or Johnson's algorithm when dealing with sparse graphs. Hence, the time complexity of algorithm APPROXRAD is equal to that of APPROXASP, proven to be $O(n^4)$ in Theorem 5.15. \square

Theorem 5.19. *Algorithm APPROXRAD computes a $(2 + \frac{\omega}{R^*})$ -approximation for the RM problem.*

Proof. Let F^* be an optimal solution for RM in the instance $G = (V, E)$ and $k \in \mathbb{N}$, hence $|F^*| \leq k$. For graph $G^* = (V, E \cup F^*)$ we denote $R^* = R_{G^*}$. Then by definition of the radius there must exist a vertex $v \in V$, where $\epsilon^*(v) = \epsilon_{G^*}(v)$, such that $\epsilon_{G^*}(v) = R^*$. Assuming algorithm APPROXSOURCE returns a $(2 + \frac{\omega}{\epsilon^*(v)})$ -approximation for the EM problem as proven by Theorem 5.12, then Lemma 5.17 shows us that for vertex v the same algorithm returns a $(2 + \frac{\omega}{R^*})$ -approximation for the RM problem.

Since the APPROXRAD algorithm applies APPROXSOURCE to all vertices in V , storing the best solution seen thus far, the result must be at least a $(2 + \frac{\omega}{R^*})$ -approximation. This approximation ratio also holds for U-RM by the same analysis. Moreover, due to the fact that $\omega = 1$ in this case, the approximation ratio is improved to $2 + \frac{1}{R^*}$. \square

5.5 Diameter

The DM problem of minimizing the diameter for a given graph $G = (V, E)$, as specified and proven to be NP-hard in Section 4.5, has been studied the most among all graph distance minimization problems. We will start by discussing the algorithm introduced by Demaine and Zadimoghaddam [10], which is simply our APPROXSOURCE algorithm with any arbitrary vertex $v \in V$ defining the source. Although initially an approximation ratio of $(4 + \epsilon)$ was proven, the bound was shortly thereafter improved by Bilò et al. to $2 + \frac{2\omega}{D^*}$ for an optimal resulting diameter of D^* . We will shortly revisit the theorem by Bilò et al, due to the fact that the eccentricity approximation ratio proof was based on this same principle.

Theorem 5.20 ([7, Theorem 1]). *Algorithm APPROXSOURCE, on an arbitrary vertex $v \in V$, computes a $2 + \frac{2\omega}{D^*}$ -approximation for DM.*

Proof. Let F^* be an optimal solution for DM in the instance $G = (V, E)$, and $k \in \mathbb{N}$, hence $|F^*| \leq k$. For graph $G^* = (V, E \cup F^*)$ we denote $D^* = D_{G^*}$ as the optimal diameter after minimization. Also, let F be the solution returned by algorithm APPROXSOURCE on vertex v , where $G' = (V, E \cup F)$.

Now consider Theorem 5.12, where a bound of $R_G(B_i) \leq D^*$ was proven for all $1 \leq i \leq k+1$. Then the situation displayed in Figure 5.4, for any $1 < i, j \leq k+1$, illustrates the bound on the diameter. Since every vertex in G' is at distance at most $2 \cdot D^* + \omega$ from v , we know that $D_{G'} \leq 2D^* + 2\omega$. Therefore, we have proven an approximation ratio of $2 + \frac{2\omega}{D^*}$.

This approximation ratio also holds for U-DM by the same analysis. Moreover, due to the fact that $\omega = 1$ in this case, the approximation ratio is improved to $2 + \frac{2}{D^*}$. \square

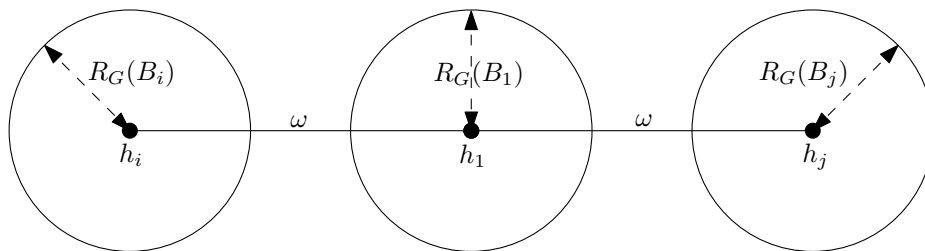


Figure 5.4: Graph illustrating the diameter bound proven in Theorem 5.20.

Theorem 5.20 has shown improving the connectivity of any arbitrary vertex in a specific way is sufficient to approximate minimizing the diameter. However, some vertices are better suited for this improvement than others, providing more benefit to the diameter due already existing connections. Hence, the idea is to improve the approximation by improving the connectivity each vertex in the graph separately and keeping track of the best resulting diameter.

However, this approach is very similar to the technique used in approximating minimization problems ASPM and RM, Algorithm 3 and Algorithm 4 respectively. Therefore we will introduce Algorithm 5 (i.e. APPROXGENERAL) below, which approximates a generalized distance-based measure M , to replace these algorithms. Note that the approximation ratios of the ASPM, RM, DM, and their unweighted variants remain unchanged, due to the algorithm having the same structure.

Algorithm 5: General approximation algorithm for distance-based measure improvement.

Input: $G = (V, E)$, graph with vertex whose measure needs improvement
 $d_G : V \times V \rightarrow \mathbb{R}_0^+$, the all-pairs distance function of graph G
 $k \in \mathbb{N}^+$, a bound on the number of added edges
 M , a distance-based measure

Output: F , a set of non-existing edges to improve the measure

APPROXGENERAL(G, d, k, M)

- (1) $F \leftarrow \{ \}, m \leftarrow \infty$
- (2) **foreach** $v \in V$
- (3) $F' \leftarrow \text{APPROXSOURCE}(G, d, v, k)$
- (4) $G' \leftarrow (V, E \cup F')$
- (5) **if** $M_{G'} < m$
- (6) $F \leftarrow F'$
- (7) $m \leftarrow M_{G'}$
- (8) **return** F

Since Algorithm 5 (i.e. APPROXGENERAL) is a generalization of Algorithm 3 and Algorithm 4, its time complexity has also changed accordingly.

Theorem 5.21. *The time complexity of algorithm APPROXGENERAL is $O(nM + n^2k)$, where $O(M)$ is the time complexity of the equally named distance-based measure.*

Proof. The structure of algorithm APPROXGENERAL has a lot in common with algorithms APPROXASP and APPROXRAD. Hence, when substituting the measure computation time complexity with $O(M)$ in Theorem 5.15 and Theorem 5.18, we get $O(nM + n^2k)$. \square

5.6 Algorithm evaluation

In the previous sections, worst case approximation bounds of the APPROXGENERAL algorithm have been proven for the three main distance-based measures, being radius, diameter and average shortest path length. However, to ensure that the algorithm performs according to expectations, we will experimentally evaluate it on some of the large randomly generated networks as seen in Chapter 3.

Of the random network generation models discussed in Section 3.1 we have selected the Barabási-Albert and Watts-Strogatz models for this evaluation due to their structure being more susceptible to improvements. Both of these models will have the same initial instance as presented in Section 3.2, i.e. 100 vertices ($n = 100$). In the Barabási-Albert model each new vertex is assigned 1 connection, whereas the Watts-Strogatz model has $\beta = 0.25$ and a mean degree of 2.

Displayed in Table 5.1 are the distance-based graph measure values after adding k edges to the graph/network using the APPROXGENERAL algorithm. Each row in the table contains the graph measures averaged over 100 randomly generated networks according to the given model.

	<i>R</i>	<i>D</i>	<i>D^{avg}</i>		<i>R</i>	<i>D</i>	<i>D^{avg}</i>
$k = 0$	7	13	5.50	$k = 0$	16	30	12.01
$k = 5$	7	11	5.11	$k = 5$	12	21	8.42
$k = 10$	6	10	4.77	$k = 10$	7	13	6.34
$k = 15$	5	9	4.38	$k = 15$	6	11	5.26
$k = 20$	4	8	4.13	$k = 20$	5	10	4.52
$k = 25$	4	7	3.89	$k = 25$	4	8	4.09

(a) Barabási-Albert model
(b) Watts-Strogatz model

Table 5.1: The values of all distance-based graph measures on randomly generated graphs before and after improvement using the APPROXGENERAL algorithm.

As can be seen in the table, all measures are indeed reduced significantly. Although no comparative results are shown here, such as the optimal measurement values (or a lower bound) after adding k edges, we can still conclude that the APPROXGENERAL algorithm is effective at reducing the distance-based measures.

Chapter 6

Conclusion

6.1 Discussion

The aim of this thesis was to find a way to quantify network robustness through the use of intuitive graph measures, and to utilize them in the process of robustness improvement. Moreover, we specifically aimed at answering the following research question.

Research question *How can we determine the robustness of a simple, connected, undirected, weighted graph, and use the resulting measures in the improvement process?*

Let us analyze the achieved results by discussing the following secondary research questions, as stated in the introduction and repeated below.

Secondary research questions

- Which graph measures are suited for measuring network robustness?
- How viable are distance-based measures in measuring the network robustness?
- What complexity class does robustness improvement, i.e. distance-based measure minimization, belong to?
- How do we go about approximating the optimal robustness improvements?
- What approximation ratios can we achieve for robustness improvements?

We started by defining some traditional graph measures from network theory in Chapter 2: radius, diameter, average shortest path length, average degree, average closeness, average vertex/edge betweenness, vertex/edge connectivity, largest connected component size, average clustering coefficient, algebraic connectivity and effective graph resistance. Each of these graph measures was attributed to one of four classes: distance, centrality, connectivity and spectral.

We continued the thesis by reviewing and comparing these measures in Chapter 3. This comparison showed that some measures are better at capturing robustness than others. More specifically, the class of connectivity measures performed poorly in the comparison due to the fact that its values did not strictly increase with the robustness. The spectral measures, although better than the connectivity measures, also did not always strictly increase. Moreover,

these measures were not as intuitive as others, since it was not always clear which properties of the graph it expressed. The centrality and distance measures performed better, average shortest path length and betweenness in particular, being strictly increasing according to an expected pattern. Out of all measures considered in the comparison, the distance-based graph measures appeared to be the most feasible. All measures in this class had similar scaling, being in the same order of magnitude, implying that they correlate well. Hence, since the measures in this class agree with each other, they present a combined method aimed at robustness analysis. Thus we concluded that distance-based measures were viable in measuring network robustness.

Afterwards we analyzed the minimization problems related to the distance-based measures in Chapter 4. Using a polynomial reduction from the NP-complete 3C problem, as defined in Definition 4.2, we were able to prove the NP-hardness of all minimization problems. This reduction was based on the concept of adding a number of specific edges aimed at solving the 3C problem. Hence, solving these minimization problems is at least as hard as solving the NP-complete 3C. Resulting from this reduction are Theorem 4.5, Theorem 4.8, Theorem 4.11, Theorem 4.14 and Theorem 4.17 respectively.

Knowing the complexity of these minimization problems, it was clear that finding a polynomial time algorithm which could compute the set of edges minimizing the measure is only possible if $P=NP$. Hence, we shifted our aim towards approximating an optimal solution to these problems. This approximation technique, as explained in Chapter 5, works by connecting a single vertex to a number of well-separated cluster heads. Using Algorithm 1 as our clustering technique, this resulted in Algorithm 2 for single source improvements, and Algorithm 5 for general graph improvements.

Approximation ratios for these algorithms were also formally proven in the same chapter, as done in Theorem 5.12, Theorem 5.13, Theorem 5.16, Theorem 5.19, and Theorem 5.20 respectively. The approximation of most measure minimization problems admitted to an approximation rate of 2, with some additional factor depending on the maximal weight ω of the added edges. The only exception was the general average shortest path, for which we proved a ratio of 4.

To summarize, in this thesis we compared different techniques of determining the robustness of a simple, connected, undirected, weighted graph. In this analysis, distance-based graph measures seemed to be the most viable. Furthermore, we presented multiple approximation algorithms aimed at minimizing these distance-based measures, thus improving robustness. Hence, our efforts have given a conclusive answer to the posed research question.

6.2 Future research

Since the results presented in this thesis are based on undirected graphs, it would be interesting to expand the analysis and some of its proofs to directed graphs. Other possibilities with respect to directed graphs would be the introduction of new graph measures, which might work better in a directed context.

Concerning the algorithms, improving on the approximation ratios is still possible. New techniques could lead to lower bounds, or even a proof stating that the optimal ratio has already been reached. As for experimental validation, it could be interesting to compare the results of our approximation algorithms to the optimal results. This would show the ratio of the algorithms in an average case, hence illustrating the tightness of our proven bound.

Another issue which could be dealt with is the introduction of potential targets through the use of our approximation algorithm. Since the algorithm adds all its k additional edges to a single vertex, this highly connected vertex suddenly becomes a target for attacks. A solution to this problem could be to call the algorithm multiple times with smaller values for k , such that the added edges are eventually distributed across many vertices. However, the optimal way to distribute the k total edges over these function calls needs to be researched further.

Finally, since we shortly discussed interdependent networks, it could be of interest to perform additional research into designing measures capable of capturing the robustness of such networks. An idea could be to use the same distance-based measure, but with the limitation that the vertices defining the endpoints of a regular distance-based measure should be part of different networks.

Bibliography

- [1] Pankaj K. Agarwal, A Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The resilience of wdm networks to probabilistic geographical failures. In *INFOCOM, 2011 Proceedings IEEE*, pages 1521–1529, April 2011.
- [2] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
- [3] R. Albert, H. Jeong, and A. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, July 2000.
- [4] J.S. Baras and P. Hovareshti. Efficient and robust communication topologies for distributed decision making in networked systems. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3751–3756, 2009.
- [5] A. Barrat, M. Barthlemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.
- [6] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish. Improving network robustness by edge modification. *Physica A: Statistical Mechanics and its Applications*, 357(3-4):593–612, 2005.
- [7] D. Bilò, L. Gualà, and G. Proietti. Improved approximability and non-approximability results for graph diameter decreasing problems. In *Mathematical Foundations of Computer Science 2010*, volume 6281 of *Lecture Notes in Computer Science*, pages 150–161. Springer-Verlag, 2010.
- [8] V. Chepoi and Y. Vaxes. Augmenting trees to meet biconnectivity and diameter constraints. *Algorithmica*, 33(2):243–262, 2002.
- [9] D.M. Cvetković, M. Doob, and H. Sachs. *Spectra of graphs: theory and application*. Pure and applied mathematics. Academic Press, 1980.
- [10] E.D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *Proceedings of the 12th Scandinavian Conference on Algorithm Theory, SWAT'10*, pages 420–431, 2010.
- [11] Peter Eades. A heuristics for graph drawing. *Congressus numerantium*, 42:146–160, 1984.

- [12] W. Ellens, F.M. Spijksma, P. Van Mieghem, A. Jamakovic, and R.E. Kooij. Effective graph resistance. *Linear algebra and its applications*, 435(10):2491–2506, 2011.
- [13] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
- [14] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.
- [15] R.W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345–, June 1962.
- [16] F. Frati, S. Gaspers, J. Gudmundsson, and L. Mathieson. Augmenting graphs to minimize the diameter. *Computing Research Repository*, abs/1309.5172, 2013.
- [17] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.
- [18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
- [19] T.F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(0):293–306, 1985.
- [20] J. M. Ho and D. T. Lee. Bounded diameter minimum spanning trees and related problems. In *Proceedings of the Fifth Annual Symposium on Computational Geometry*, SCG '89, pages 276–282, 1989.
- [21] D.B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1):1–13, January 1977.
- [22] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.
- [23] S. Kapoor and M. Sarwat. Bounded-diameter minimum-cost graph problems. *Theory of Computing Systems*, 41(4):779–794, 2007.
- [24] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.
- [25] Y. Koç, T. Verma, N.A.M. Araújo, and M. Warnier. Matcasc: A tool to analyse cascading line outages in power grids. *Computing Research Repository (CoRR)*, abs/1308.0174, 2013.
- [26] Y. Koç, M. Warnier, R.E. Kooij, and F.M.T. Brazier. A robustness metric for cascading failures by targeted attacks in power networks. In *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, pages 48–53, April 2013.
- [27] Y. Koç, M. Warnier, P. Van Mieghem, R.E. Kooij, and F.M.T. Brazier. The impact of the topology on cascading failures in a power grid model. *Physica A: Statistical Mechanics and its Applications*, 402:169–179, 2014.

- [28] Chung-Lun Li, S.Thomas McCormick, and David Simchi-Levi. On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems. *Operations Research Letters*, 11(5):303–308, 1992.
- [29] M. Manzano, E. Calle, and D. Harle. Quantitative and qualitative network robustness analysis under different multiple failure scenarios. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, pages 1–7, October 2011.
- [30] M. Manzano, E. Calle, V. Torres-Padrosa, J. Segovia, and D. Harle. Endurance: A new robustness measure for complex networks under multiple failure scenarios. *Computer Networks*, 57(17):3641–3653, December 2013.
- [31] B. Meyer. Self-organizing graphs a neural network perspective of graph layout. In *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 246–262. Springer Berlin Heidelberg, 1998.
- [32] A. Meyerson and B. Tagiku. Minimizing average shortest path distances via shortcut edge addition. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 272–285, 2009.
- [33] B. Mohar. Laplace eigenvalues of graphs - a survey. *Discrete Mathematics*, 109(1–3):171–183, 1992.
- [34] K. Okamoto, W. Chen, and X. Li. Ranking of closeness centrality for large-scale social networks. In *Proceedings of the 2nd Annual International Workshop on Frontiers in Algorithmics*, pages 186–195, 2008.
- [35] A.A. Schoone, H.L. Bodlaender, and J. Van Leeuwen. Diameter increase caused by edge deletion. *Journal of Graph Theory*, 11(3):409–427, 1987.
- [36] Y. Singer. Dynamic measure of network robustness. In *Electrical and Electronics Engineers in Israel, 2006 IEEE 24th Convention of*, pages 366–370, November 2006.
- [37] A. Sydney, C. Scoglio, P. Schumm, and R. E. Kooij. Elasticity: Topological characterization of robustness in complex networks. In *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems*, pages 19:1–19:8, 2008.
- [38] H. Taichi. Robustness and directed structures in ecological flow networks. *Computing Research Repository (CoRR)*, abs/1306.6129, June 2013.
- [39] P. Van Mieghem, C. Doerr, H. Wang, J. Martin Hernandez, D. Hutchison, M. Karaliopoulos, and R. E. Kooij. A framework for computing topological network robustness. *Submitted for publication*, 2010.
- [40] V.V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.
- [41] H. Wang, R.E. Kooij, and P. Van Mieghem. Graphs with given diameter maximizing the algebraic connectivity. *Linear Algebra and its Applications*, 433(1112):1889–1908, 2010.

- [42] X. Wang, E. Pournaras, R.E. Kooij, and P. Van Mieghem. Improving robustness of complex networks via the effective graph resistance. *Submitted for publication*, 2014.
- [43] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
- [44] J. Wu, M. Barahona, Y. Tan, and H. Deng. Spectral measure of structural robustness in complex networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(6):1244–1252, November 2011.
- [45] M. Youssef, C. Scoglio, and S. Pahwa. Robustness measure for power grids with respect to cascading failures. In *Proceedings of the 2011 International Workshop on Modeling, Analysis, and Control of Complex Networks, Cnet '11*, pages 45–49, 2011.