Estimating the Vulnerability of Ship Distributed System Topologies

S.P. van Leeuwen

SDPO.17.020.m

TU Delft

Defensie Materieel Organisatie
Ministerie van Defensie

Estimating the Vulnerability of Ship Distributed System Topologies

by

S.P. van Leeuwen

Performed at

Defence Materiel Organisation (DMO)

to obtain the degree of Master of Science in Marine Technology in the specialization of Marine Engineering
at the Delft University of Technology,
to be defended publicly on Tuesday August 22, 2017 at 09:00 AM.

Company supervisor:	Dr. Ir. E.A.E. Duchateau	
E-mail:	EAE.Duchateau@mindef.nl	
Project duration:	Oct 1, 2016 – August 22, 2017	
Thesis committee:	Ir. K. Visser,	TU Delft, chairman
	Ir. P. de Vos,	TU Delft, supervisor
	Dr. Ir. E.A.E. Duchateau,	DMO, supervisor
	Dr. R.R. Negenborn,	TU Delft
Student number:	4090349	
Author contact e-mail:	Simonv.leeuwen@hotmail.com	

TU Delft

Defensie Materieel Organisatie
Ministerie van Defensie

Preface

Before you lies the thesis "Estimating the vulnerability of ship distributed system topologies" in which a method is introduced to determine the vulnerability of system topologies during an early design stage. This thesis was written in correspondence with the requirements of the MSc graduation activities for the Delft University of Technology, faculty of Mechanical, Maritime, and Materials Engineering. The MSc graduation project was executed at the Defence Materiel Organisation in The Hague during the period of 2016-2017 by S.P. van Leeuwen.

This project results from the request of P. de Vos whom required the identification of a method to sort topologies based on their performance for this PhD study.

During the project, a method was created to analyse system topologies based on their performance with respect to their vulnerability. The method is applicable to a wide range of problems ranging from finding the best balance within a distribution network to assessing the topology of an entire vessel. The method works by using network theory to identify connections and check whether the energy requirements of components are met. It is my hope that the method results in the realization of better balanced design solutions for the system design of vessels.

I would like to thank the Defence Materiel Organisation for the possibility to do my MSc research with them. I want to express my gratitude to everyone at DMO for their support and feedback, most of all my supervisor Dr. Ir. E.A.E. Duchateau.

In addition I would also like to thank my supervisor P. de Vos for his support and help during this thesis. As well as the support with gaining an understanding of his tool, SDS-ATG.

S.P. van Leeuwen
Delft, August 2017

Disclaimer

The work presented in this Delft University of Technology MSc thesis has been performed at the Defence Materiel Organisation (DMO) of the Dutch Ministry of Defence. However, it is important to note that the work, the frigate design and system concepts, and the test-cases with results, as presented in this thesis, *do not* reflect any Royal Netherlands Navy ships or any planned or ongoing procurement programs at DMO. All presented ship and system concepts where chosen and altered in such a way that they are realistic, yet not representative of any (future) Royal Netherlands Navy ships.

Summary

During the ongoing PhD research of de Vos [2017] the "ship distributed systems automatic topology generator" (SDS-ATG) tool was created. This tool combines the fundamentals of network theory and marine engineering to automatically generate ship distributed system topologies. An optimization algorithm is used to optimize topologies with respect to system performance and cost. The performance objective can be related to the robustness of the system topology (e.g. reliability or in the case of naval ships to the broader term vulnerability).

This MSc thesis focusses on the investigation and development of a better vulnerability assessment method, based on network theory, that can be used to assess the distributed system topologies created by SDS-ATG. The assessment method will be used to better define the performance of (automatically generated) system topologies in early stage design.

The first step in developing the new method is to explore the options that are available in the literature with respect to vulnerability assessment of networks. These included typical network theory metrics (e.g. characteristic length, edge betweenness, and degree centrality) as well as the more advanced maxflow network analysis. These existing methods and metrics are mostly based on the characteristics of the network and give a prediction of the performance of one network with respect to another. None of these methods and metrics was deemed feasible, as they either were not fit to be used as an objective function for the optimization tool, or they did not properly take into account the marine engineering system present within a distributed system topology. Nonetheless, the exploration of these available methods did provide insight into several aspects which should be taken into account. These are:

1. A generalization of the network topologies cannot be used, as the system logic does not allow it.
2. Every node should be evaluated individually, because of the interdependencies between different systems within the distributed system networks.
3. Deterministic methods should be used which assess vulnerability based on networks in which damage is simulated.

Besides the methods and metrics in literature, an assessment method was already available in the SDS-ATG tool. The maximum HUB flow method takes some system logic into account, but the results of this method were not deemed desirable.

Using the insight gained from examining several existing methods and metrics, a new method was developed. It was established that a prediction method, such as available from literature, was not desirable. Hence, the failures of the topologies had to be modelled in order to assess the reaction of the system topology to these failures.

The developed method is separated into two parts which both use the modelled failures. The first part is used to give a good prediction of the vulnerability of a system topology using the connections that are available within the topology after failure. The second part uses the still connected topologies of the first part to assess whether the user defined critical components can still be supplied with enough resources through the damaged network topology. With the knowledge of the availability of supply to components, a accurate vulnerability assessment can be made.

With a new method defined to determine the vulnerability of a system topology, the next step is to verify and validate the method and to look into the results of the method. For the verification, a smaller multi-layered network was used that consists of all of the situations that can occur within a topology based on failure. The vulnerability of these networks was determined by the method as well as calculated by hand. By comparing the two results, it was shown that the method is able to accurately predict the vulnerability of different network topologies.

The results of the method showed how the vulnerability is affected by the composition of the distribution networks. By comparing different distribution networks, it can be seen that after a certain point, the vulnerability increases in small steps while the total size of the network expands rapidly.

Besides smaller networks, the method was also used to analyse a concept system design of a notional frigate. The results of this test-case showed that by altering the distributed system topology reduction of both the size and vulnerability is possible for the concept frigate system.

To conclude, the developed vulnerability assessment method is seen to greatly increase the capability of the SDS-ATG tool as an early stage system design tool.

Contents

1 Introduction

All ships rely on a large amount of systems on board to function. These systems often belong to a variety of system groups, such as: propulsion systems, distribution systems, hotel systems and specialized systems. The first three groups can be found on every vessel, while the last group is mainly reserved for complex vessels. A non-complex or 'normal' vessel can be a container vessel as shown in figure 1.1a opposed to a complex vessel such as the naval vessel in figure 1.1b or a pipe laying vessel as seen in figure 1.1c. The three vessels shown in figure 1.1 have been built for entirely different missions. These missions support a variety of tasks that the vessel needs to be able to carry out. Due to these different missions, the layout of the ships is different. Considering the system density or the amount of volume the systems occupy against the entire volume of the vessel, it can be seen that the container vessel has a much lower system density due to its mission specific layout. The two complex vessels have a higher system density due to a large amount of mission specific or specialized equipment. Hence, from a system design point of view these vessels can be considered to be more complex, than for instance a container vessel or bulk carrier.

(a) Container vessel

(b) Naval vessel

(c) Pipe laying vessel

Figure 1.1: Different vessels

The past decade, new tools have been developed that change the workload of the engineers designing these vessels. During the preliminary design phase, the workload has shifted from the actual design of concepts (drawing and calculations) to the programming of models and the understanding of the generated data (e.g.

to give insight into the design problem). The tools that have made this change possible can generate many different concept designs using the information input of the engineer. One of the tools that has been developed is the Ship Distributed Systems Automatic Topology Generator or SDS-ATG [de Vos, 2014]. The SDS-ATG tool allows for a rapid generation and evaluation of many different conceptual system topologies for (combined) ship distribution systems (e.g. the electrical distribution network and/or the chilled water distribution network).

The goal of this MSc thesis is to define a method to assess the properties of the system topologies[1] generated by SDS-ATG. The properties that will be included are cost, operability and vulnerability. Cost and operability can be related to the size of the system topology using the assumptions given in chapter 3. The vulnerability is predicted using a method that simulates internal and external disturbances acting on the systems. With internal referring to everything that could happen to systems that can be expected aboard of a vessel, including failure of components. While external disturbances include sustaining damage (e.g. caused by a collision or a weapon impact in case of a naval vessel).

1.1. Ship design process

The general ship design process was mentioned above to be mostly the same for both normal and complex vessels. This section will give an introduction into this general ship design process. Many references discuss the typical ship design process in detail [Andrews, 1994, Brown, 1991, Erikstad, 1999] as shown by Duchateau [2016]. The terminology between the references differs, but they all identify three main phases within the design process. These are:

1. *Preliminary design*
   During this 'early stage design' different solutions are explored that can fulfil the needs of the customer while keeping to the available budget. These concepts are used to assess the technical impact and cost implications of the requirements. This insight is then used to reassess the requirements together with the customer in search of a desirable balance design solution. The balanced design solution(s) will then be worked out in greater detail, so that the final solution can be chosen. The final solution will be used in the next phase of the process

2. *Contract design*
   A design based on the chosen concept, but at a higher level of detail. This concept is used to assess the detailed costs and possibility for production.

3. *Detailed design*
   The final design of a vessel is made in the highest level of detail, as it is used to produce the vessel. This phase involves the generation of detailed drawings describing every part of the vessel. Often, this phase overlaps with the actual construction of the vessel to save time.

The above three phases show that many of the decisions concerning the design of a vessel are made during the preliminary design phase. This means that whenever a decision is recalled during a later phase of the design, a lot of time and resources are needed to correct the contract/detailed design. It is thus important to include as much information within the preliminary design as possible without the need for more time and/or resources.

1.2. System design

The previous section gave a short introduction in the ship design process. Using the provided information, it is possible to give system design a place within this process. The easiest way to place system design is to mention that it is included in every phase of the ship design process. However, it is the level of detail of the system design within each phase that is of interest here.

The preliminary design phase holds the key to determine a balanced design solution. It is therefore undesirable that the system design during this phase is at a low level of detail, yet this is often the case because of the earlier mentioned constraints in resources. In most cases, the most important systems are considered, but it is assumed that they will fit within the design as to shift the actual system design to a later phase. The system design during the first phase is thus mainly concerned with different concepts to be used and not

[1] In the context of this thesis, a system topology is a representation of the components and their connections within a distributed system, see chapter 3 for more information.

so much with placement (except for the main components of a vessel such as: drive-train and specialized systems) and detailed assessment of these concepts. This focus during the preliminary design can be justified as broadening the scope during this phase would mean an exponential increase in considered systems and interactions between systems. With an exponential increase in systems to consider, the time needed to investigate different concepts would also increase by a substantial amount.

When a design solution has been chosen, a more detailed design is made resulting in a more detailed systems design. The final two phases realize a transformation from basic systems to a fully functional system design. The scope increase during this process, slowly increases the total system design as all system tasks are filled. The last systems to be added are in general the distribution systems, leaving the placement of pipes and cables to the last moment of the design coinciding with the actual construction of the vessel.

The above approach to system design is flawed. The assumption that all systems will fit is not always valid. For normal vessels such as the container vessel, the assumption will most likely hold true. However, complex vessels have a system design that dictates the size and the performance of the vessel. For example, the placement of the distributed systems and cables/pipes on board of a naval vessel greatly dictate how the vessel is influenced by damage. A problem that arises from this method of design is the need of a redesign during a later phase, because the general arrangement is not correct. The needed redesign increases the cost of the vessel, which could be significant when construction of the ship has already begun. It is thus the layout, size and weight of the different systems and their components that become an important factor that drive the size and general arrangement of the vessel. Hence, it is needed to consider a more detailed system design during the preliminary design phase [van Oers, 2011].

1.3. Redundancy

When creating a system design it is important that a vessel can reliably complete its mission, whether it is moving containers from destination A to destination B (container vessel) or dealing with military threats (frigate). One of the easiest methods to ensure this is by increasing the redundancy in the system design. Redundancy as referred to within engineering is:

> *"The duplication of critical components or functions of a system with the intention of increasing reliability [or survivability] of the system." Wikipedia [2017]*

The above statement shows that redundancy can be used in two types of situations: reliability and survivability. The reliability of a vessel will be explored first, followed by the survivability.

1.3.1. Reliability

The reliability shows how high the probability of success is for a certain defined mission or goal. For vessels, it would mean the probability of the vessel to complete its mission. From a system design point of view, the reliability can be estimated and measures can be taken. The impact of the measures taken differ per vessel, as both the system designs and the mission differ.

Due to the difference in impact of measures, the amount of work needed to gain the exact same level of reliability is different for every vessel. The 'normal' vessels such as a container vessel have a low system density resulting in less work to get a redundant system. However, for these 'normal' vessels the cost of getting redundancy is in most cases too high respective to the increase in reliability. For instance, container ships mostly have only a single propulsion engine as the efficiency gain of a single engine outweighs the benefit of extra redundancy of two main propulsion engines.

Complex vessels on the other hand have a high system density. The larger number of systems, creates a challenge with respect to redundancy. It is for these ships that a reliability increase is worth the cost of a redundant system. Most of the complex vessels only have influences on their systems from within the vessel apart from some weather influenced external influences. These influences can lead to failure, but the time between failures is long, and it can be that regular maintenance works as good as increased redundancy of the systems.

1.3.2. Survivability

A special case of reliability is the survivability, it is connected to a certain group of vessels, the naval vessels. Survivability is an extension of the reliability as shown above, it determines the chance to continue a mission after handling threats. Survivability is a combination of three aspects of a vessel resisting an external (weapon/military) threat. These aspects are:

1. *Susceptibility*
   The first reaction to a threat is prevention of the threat. This aspect handles everything from evading the threat before it happens (e.g. by applying stealth technology) to destroying the threat before it can have an impact on the vessel (e.g. by so-called countermeasures).

2. *Vulnerability*
   When a threat damages the vessel, the vulnerability handles the impact of the damage done to the vessel. This can be either by resisting the effects of the threat (redundancy) or by minimizing the damage done due to the threat (e.g. by applying measures such as blast bulkheads or armour plating)

3. *Recoverability*
   When damage has occurred to the vessel, the response of the vessel and the crew to the damage, such as: minimizing the effect (fire fighting) or restoring capability (repairs), can be grouped under the recoverability.

From the above three subgroups, the second subgroup, the vulnerability is the focus with respect to redundancy and system design. To get a better understanding of the vulnerability it is necessary to give a clear definition. NATO defines vulnerability as [Parent, 2015]

"Vulnerability is the extent of degradation of a system after having been subjected to combat threat(s), that is, the degree of mission impairment as a result of sustaining finite levels of damage caused by weapon hits."

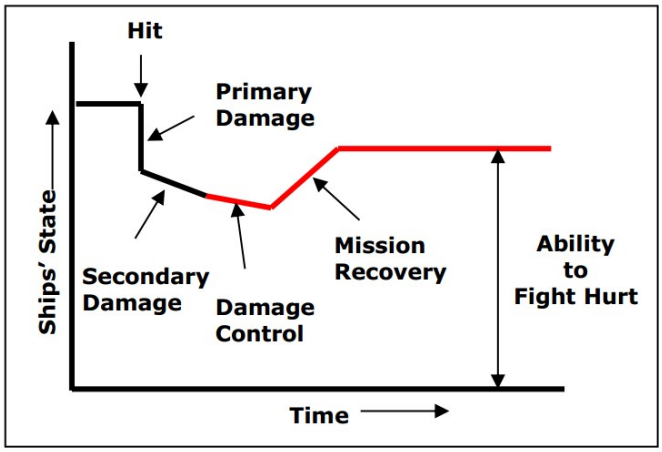


Figure 1.2: Survivability of a vessel including recoverability [Parent, 2015]

To illustrate this statement, figure 1.2 is shown. This figure shows how damage can be seen over time after a hit from an external threat. The first part of the graph show the functionality of the ship before a hit. After the hit, there is an immediate negative response, as the primary damage is done. After impact, there is a chance of fire, flooding or other faults of the vessel which lead to the secondary damage of the vessel. At this point damage control is activated as part of the recoverability. During damage control, damage is still done to the system, it is thus that the vulnerability within this figure is until the mission recovery (e.g. until the degradation of the ships state has stopped).

1.4. PACKING, topology generation and routing

The exploration of multiple solutions is the key to finding a balanced solution during the preliminary design stage. The amount of possible solutions that are explored is however also based on the budget available for the project. For many years, the different solutions were found using the experience and knowledge of engineers and input of the customer. Since exploring the solutions was a manual job, the process was time consuming. This leads to a few problems. Due to the time-consuming nature of the process it is easier to use solution well known to the engineer instead of exploring a new untested solution. Also, a limited number of solutions is explored due to a limited amount of time available.

During the last decade, research has focussed on automating parts of the preliminary ship design phase. This research is based on the solutions needed during the preliminary design phase. From the research a tool was created, the Packing-approach [van Oers, 2011]. This tool can automatically generate viable ship layouts based on user input building blocks. These building blocks can model; cabins, machinery, stores, hallways, etc.

Packing was improved upon by enabling a feedback loop between the program and the user [Duchateau, 2016]. This feedback loop enables the user to define criteria the vessel should follow. These criteria are then used to find the solutions that match these criteria from the entire solution set of the program. After finding these solutions the program can continue to find new solutions with the new criteria.

The following quote was taken from the PhD thesis of van Oers [2011]:
"Developing the parametric model can only start after the ship's systems and the design requirements are available. Designing the systems and deriving requirements are both important (due to their impact on the resulting ship design), and time-consuming. Hence, support for this part of the design process is essential."

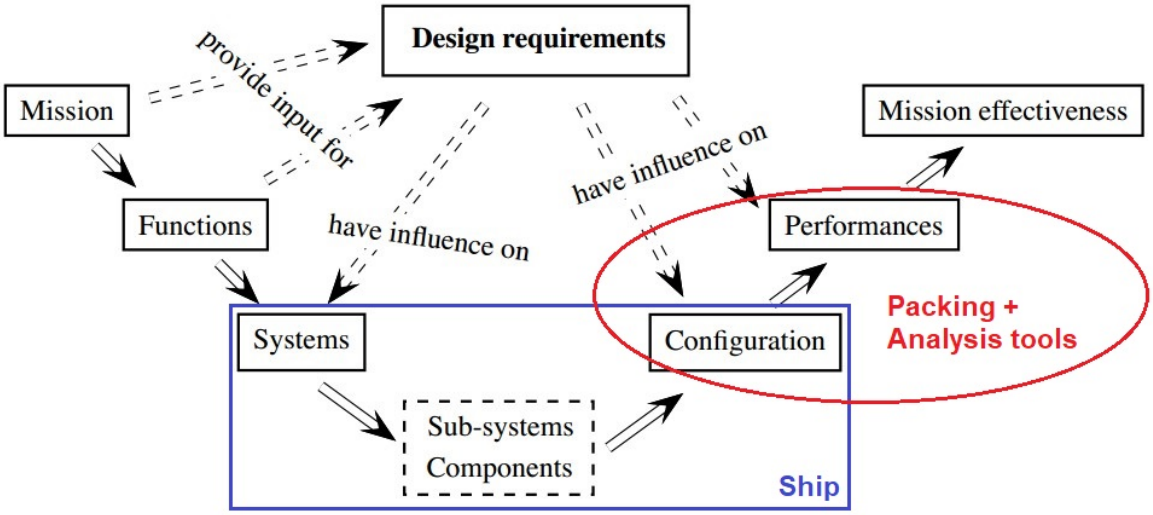


Figure 1.3: Influence of design requirements [van Oers, 2011]

The statement together with figure 1.3 identifies the importance of the system design when creating a model with, for instance; the Packing-approach. Figure 1.3 shows the influence of both the design requirements and the system on the overall configuration of the vessel. It is this configuration that is created with tools such as the Packing-approach. Following the recommendations of van Oers, the PhD thesis of de Vos et al. [2017] was created introducing the SDS-ATG (Ship Distributed Systems Automatic Topology Generator).

As mentioned before, the SDS-ATG generates system topologies. These topologies can be generated with a variable level of detail based on the amount of information available. During the preliminary design phase the tool could generate the most important distribution networks based on the information of key components. While later design phases may demand the generation of small scale or more specialized distribution networks. Beside the ability to adjust the level of detail, the tool also provides the ability to add components at will. Using this feature allows the designer/engineer to add or dismiss components, and to investigate the impact this could have on the system topology properties. This can lead to an investigation whether a large component can be best split up into two smaller components or the other way around. It is also possible to determine whether a extra distribution component is necessary.

Finally, when a system topology has been chosen it can be routed through the layout of the vessel as generated by the Packing-approach [van Oers et al., 2012]. The topology routing creates the opportunity to determine the geometric properties of the topologies. These properties can be used to assess the topology at a higher level of detail. For instance, a robust topology can have mayor redundancy issues when the same topology is routed throughout the vessel. Important components could be placed within the same compartments or those redundant connections could be routed through the same space. Thus, it is important to route the system topologies to be able to choose the best solution.

1.5. Relevance of the assessment method

Past advances in the automatic generation of multiple concepts have made it clear that system design should be considered during the preliminary design phase. Due to these developments, the SDS-ATG was created to give an approach to distribution system concept design that was not present before. This new approach generates many varying system topologies. Before anyone can choose the best topology for a balanced solution, it is necessary to assess the performance of different system topologies.

The method described in this thesis can assess the vulnerability of a system topology. Since the concept of generating system topologies is new to ship design, no method has been described for this specific instance. In literature however, many other metrics and methods can be found for similar design problems and networks. Some of these are insufficient to handle the information within the system topology, such as the system logic. System logic is the correlation between different distribution systems and their respective components. While others can handle the system logic, but are assessing a undamaged system resulting in a prediction of the vulnerability to be expected, rather than the actual vulnerability of a system.

This research will focus on developing a tool that circumvents the problems met when using the current metrics and methods. Using the information available of the system topologies, the new method will be able to create new topologies that, based on its requirements, can be used to select the 'best' topology. This topology will be selected based on both the cost of the topology and the vulnerability of the topology. The last will be based on the capability of the topology to work even when the system has taken damage.

2

Problem analysis

The emergence of new tools to generate concept designs of vessels during the preliminary design phase, has shown the need to re-evaluate the role of system design in earlier design phases. This shift is generally invoked by the need to generate more information about the system placement and their connections in an earlier stage of the design. Regarding the connections, a new tool by the name of SDS-ATG has been introduced to automatically generate different system topologies. These generated system topologies need to be assessed to help the designer determine a balanced topology. Although different methods and metrics exist to help with the assessment, a specialized method for the assessment of marine system topologies made with network theory is not yet present. This chapter explores the steps that should be taken and the problems that should be solved to create such an specialized method.

2.1. Research questions

Based on the problem as described above, namely the need for a specialized method to assess system topologies made using network theory, the following main research question (MRQ) can be defined.

MRQ *How to assess the vulnerability of a conceptual design for distribution systems generated by SDS-ATG (which is based on network theory)?*

Since the answer to this main question is not easily solved, a few smaller research questions can be formulated to help find an answer to the main question. These research questions (RQ) are:

RQ1 *Which problems result from using network theory to describe system topologies, and how could these problems be solved?*
To create a network theory driven system topology, the real topology needs to be transformed. This transformation can only be accomplished with the use of assumptions and simplifications. This research question evaluates the resulting problems and in addition, this might lead to possible solutions to these problems.

RQ2 *Which generic methods and metrics exist to assess the vulnerability of networks in network theory?*
Using the generic methods found from the network theory, a first assessment of the problem can be completed. This assessment will give a better insight in the problems regarding the existing metrics and the improvements that should be accomplished for a new method.

RQ3 *What is missing from the custom evaluation method of the SDS-ATG tool?*
The SDS-ATG tool has a simplified built-in vulnerability prediction method. This method should be investigated to determine if problems exist and how these problems can be solved.

RQ4 *How is vulnerability defined in the case of distribution systems on board vessels?*
Vulnerability is a concept that needs to be considered before applying it to the marine system topologies. In literature, different definitions can be found to describe the vulnerability (see RQ4.1). It will be necessary to determine how the vulnerability should be defined within the confines of this thesis. To find a suitable definition for the vulnerability, this problem is separated into three sub-research questions.

RQ4.1 *How should vulnerability be interpreted?*
Vulnerability is an open concept with many different descriptions and interpretations of how it should be defined. For instance, Dictionary.com [2017] defines the vulnerability as: *"to be capable of or susceptible to being wounded or hurt, as by a weapon."* The first step to determine the vulnerability should thus be the identification of the definition to be used.

RQ4.2 *At what level of detail should the vulnerability be determined?*
For a system topology, several different levels of detail can be considered when predicting the vulnerability of the system. These levels of detail are highly dependant on the amount of information available to the engineer for the analysis and the depth needed. The difference in level of detail could be: a critical users needs to be supplied with a certain resource (electrical power, chilled water, data, etc.) or a critical users needs to be supplied with a certain amount of a certain resource. In one instance the vulnerability is based on the establishment of a connection between a supplier and a critical user. The other situation needs the same connection but a certain amount of the resource needs to be available through that connection.

RQ4.3 *Should damage be inflicted to the network to predict the vulnerability?*
The vulnerability assessment method could use an intact network to predict the vulnerability. However, it is also a possibility to use the network to create damaged copies that are then used to calculate the vulnerability. A choice should be made with respect to the approach that should be taken to determine the vulnerability.

With the above four sub-questions, a roadmap to reach an answer to the main question has been created. The layout of this thesis is based on this roadmap. First off, chapter 3 will describe the SDS-ATG tool and will give an answer to RQ1. Chapter 4 will show the different metrics and methods to predict/estimate the vulnerability of the system topologies. Section 4.2 will show the metrics found from literature and will answer RQ2. The next section of the chapter, section 4.3, will describe the metric used in the SDS-ATG tool. Within this description, an answer to RQ3 can be found. The developed method is discussed in section 4.4 providing the answers to RQ4 and its sub-questions. With the answering of all the research questions, the main research question can be answered concluding this thesis.

2.2. Research goal

From the above questions and the introduction, it should be clear that the goal of this MSc thesis is to develop the right vulnerability assessment method for the SDS-ATG tool. This method should be used to predict the vulnerability of the generated system topologies, in order to create an basis for the engineer to choose a system topology that is a balanced design solution.

The assessment method should be created in a manner consistent with the SDS-ATG tool in order to ensure its integration into this tool. The information provided by this tool will be used as input into the assessment method. This means that the first iterations or design problems that the assessment should be capable to handle are of a low level of detail. However, later iterations may need a higher level of detail, thus the level of detail needed to use the assessment method needs to be scalable. This scalability must be for simple to complex systems and the other way around.

2.3. Literature study

This section will be used to discuss most of the literature that has been collected during the research period. These sources are the publications that have been written on the many subjects that need to be explained to answer the main question, some of these subjects are; ship design, system design, vulnerability of ships/systems/networks and network theory. To keep this section readable, it has been divided into several subsections. The foundation of the research consisting of previous research, both direct as indirect will be the first subsection. Secondly the sources on network theory will be cited. Finally, the sources that describe the vulnerability of networks will be shown.

2.3.1. Previous research

The literature found on ship design is abundant, even more so when considering the problems of vulnerability, redundancy and resilience. The important sources that create the foundation of this research are discussed here. The directly involved literature is described first, showing the reasons why this research came

into existence. The indirect literature, shows notable literature that covers almost the same direction as this thesis.

Directly connected

The first sources to be evaluated consist of the publications created by de Vos. The tool SDS-ATG was proposed in a paper in 2014 as an outcome of the PhD thesis of de Vos [2017]. Although currently a work in progress, this PhD thesis has created the problem studied during this MSc thesis. Beside the PhD thesis, a new paper will also be published in the near future [de Vos et al., 2017]. The research done by de Vos introduced the SDS-ATG tool, but besides this tool it also shows a method that can be used for the vulnerability assessment of the generated topologies. This maximum HUB flow assessment method will be discussed in section 4.3.

Before de Vos began his work on the SDS-ATG tool, the Packing-approach was proposed in the PhD thesis of van Oers [2011]. This approach gave a designer the ability to model a vessel using user input building blocks. The program would then use these building blocks to create general arrangements that are solutions to the constraints given to the design. The PhD thesis of Duchateau [2016] improved upon the Packing-approach by enabling the designer to use a feedback loop. This feedback loop is used to define certain criteria that can be used to isolate desired solutions from the solution set. Using these solutions, the Packing-approach can generate new solutions matching the additional added criteria.

The topologies generated by the SDS-ATG tool can be routed through a vessel generated by the Packing-approach using the methods as described by van Oers et al. [2012]. This paper shows the possibility to route distribution systems in generated designs thus enabling the routing of generated system topologies. The method created during this thesis could be used to create a link between the topology generation and routing. The routing method is currently under development at DMO, but once the interaction is possible more diverse solution to the topology can be explored.

The last source used to inspire this research is the thesis by van Ingen [2011]. This MSc thesis proposes a method to reduce the vulnerability of naval vessels by determining the failure of systems after damage of the vessel. The method explained in this thesis is also based on the work of van Oers and shows how the routed distribution systems could be assessed. Van Ingen uses prime numbers to determine whether components within systems are still linked. When they are, he uses a linear optimization to consider the supply and demand of these nodes. Although his approach inspired the approach used in this thesis, it still had some troubles. First, the method by van Ingen is not considering distribution nodes such as switchboards. Secondly, generating larger distribution systems increases calculations times considerably. These problems could most likely be improved upon using network theory.

Indirectly connected

Only a single publication made the list of indirectly connected work, this publication is the PhD thesis of Rigterink [2014]. This thesis is concerned with methods for analysing early stage naval distributed system designs, which also fits within the scope of the current research. The difference between this thesis and that of Rigterink, however is that Rigterink uses detailed distributed system designs to prove how some of the metrics could work. The system designs on board vessels belong to either the complicated or complex networks as they are described within the PhD thesis of Rigterink, the network scale (simple to random) can be seen in section 2.3.2. Although this might be correct for the original system design, the system designs used during this research have less detail placing them into the simple to complicated networks range. The high amount of detail used by Rigterink also excludes use of his metrics and methods as the generated topologies lack the level of detail necessary.

2.3.2. Network theory

Since this thesis is based on the SDS-ATG tool which uses network theory to describe system topologies, several sources have been sought to elaborate on this theory. The first source is a book by Hillier and Lieberman [2010], *Introduction to Operations Research*. This book introduces the network theory and some of its basic concepts for simple networks. These concepts are the simplex method as well as the linear programming problems arising from simple networks.

With the simple networks covered, the next publications deal with the more complex networks. Cohen and Havlin [2010] wrote the book: *Complex networks - structure, robustness and function*. This book together with the publication by Newman [2003] on the *Structure and function of complex networks* gave enough insight on how to quantify the networks used within the network theory.

To be able to understand the placement of the networks used later in this thesis, it is necessary to identify the different levels of complexity in networks. The differentiation of networks is done within the confines of the complex networks theory, but based on the findings of Rigterink [2014]. The four groups of complexity are (with increasing complexity): simple, complicated, complex and random. The definition of these four groups is as follows:

1. *Simple:* A simple network models a system of very few parts that has a behaviour based on very basic laws or rules. An example of such a network is the transportation of goods between two points using a conveyor belt, as shown in figure 2.1. It can be easily understood that for such a system the input is equal to the output.

Figure 2.1: Simple conveyor belt

2. *Complicated:* A step up from simple networks are the complicated networks. These networks model systems that consist of many parts that behave according to very basic laws or rules. The complicated systems are mostly a combination of many simple systems. A complicated system is shown in figure 2.2, this complicated system is built from many simple conveyor belt systems.

Figure 2.2: Complicated conveyor belt system

3. *Complex:* From complicated networks to complex networks is another step up. The systems modelled by complex networks consists of many subsystems that have a high level of interaction with each other. The subsystems can be composed of smaller sub-subsystems resulting in a grand collective behaviour that is determined by the behaviour of its individual parts. For complex systems, the specific interactions between subsystems may be hard to model, however the overall behaviour can be modelled. An example of this complex system is the behaviour of the conveyor belt system of figure 2.2 when we consider that it is a subsystem of the cargo handling plant of an airport. However, each conveyor belt is propelled by a sub-subsystem, electric motor(s). Also at the beginning and end of the subsystem conveyor belt system there is cargo handling by people. This overall system becomes more and more complicated when more parts get involved. Although it is hard to model the system with every part included, the overall distribution of cargo can be modelled.

4. *Random:* The final network group is random. These are the networks that model systems with subsystems that interact with each other in a manner that makes it impossible to model the overall behaviour of the system.

From the identification of the four groups, the systems on board of vessels can be grouped under the complex networks. With enough simplifications, the entire system can be modelled. When breaking the overall system into smaller subsystems, it can also be stated that these layers behave as complicated networks, as long as complex systems are modelled as simple point systems.

2.3.3. Vulnerability identifiers of different networks

The final group of publications to be discussed is concerned with the vulnerability of networks in different fields. Four main groups concerning these publications can be identified. These groups are concerned with: power electric networks, information networks, infrastructure and theoretical work on networks. Each of these groups will be elaborate on.

Power electric networks

A large amount of the distribution networks on board of vessels are concerned with the distribution of electric energy. It was assumed that the power electric networks found in literature would present practical problems with solution that could be used for the system topologies. Dwivedi et al. [2010] tried to analyse the vulnerability of a power network using a maximum flow centrality metric. Although the metric itself is not used during this thesis, it is the indication to look at power networks as directed networks that has been utilized for the distribution systems. Guohua et al. [2008] also described how an bulk power grid should not be modelled as an undirected network, thus providing extra strength to the findings of Dwivedi et al.. Xu et al. [2009] described different topological characteristics to describe power grids. From these characteristics, the characteristic length L was used in section 4.2.1.

Information networks

Yang et al. [2016] looked at the vulnerability assessment of the shipboard electrical power information network. Although the findings of this paper are not put into practical use within this thesis, this paper gave more insight in the simplifications made during this thesis especially with regards to the information distribution network. This paper is also a good reminder that taking a scope that is too big will lead to a network to complicated to reach the objective needed by assessing the distribution systems.

Infrastructure

The work done on object-oriented modelling of critical infrastructures by Eusgeld et al. [2009] helped exploring the possibility to do just this for the distribution systems. Although the manner of object-oriented modelling used by Eusgeld et al. is not translated directly into this thesis, the object-oriented modelling is being used by the developed method. Eusgeld et al. proposed that in an infrastructure problem that is too difficult to model as a whole it would be easier to find a weak link using complex network characteristics and to model these weak links in more detail.

The object-oriented modelling method together with the definitions given for the different complexities present in networks in section 2.3.2. gave the idea of layering the model. So the entire system topology was cut into smaller distribution networks to create a simpler model to observe.

Theoretical work on networks

The final publications are concerned with theoretical work into networks. Mishkovski et al. [2010] uses a normalized average edge betweenness to determine the vulnerability of complex networks. This use of a this normalized edge betweenness led to choosing the edge betweenness as one of the metrics to be taken into account.

Since the papers describing the metrics are based on undirected networks, it was necessary to use the work of Boccaletti et al. [2007], Mao and Zhang [2013] and Albert and Barabási [2002] to form a translation between the undirected metrics and the distribution system case.

The last publication is of Holme et al. [2002] which discusses the attack vulnerability of complex networks. Their strategy was to develop different attack strategies to find the strength of a network. In light of this thesis and the randomness of actual failure within a system, it has been chosen to use random hits/failures instead.

3

Generating distributed topologies

The past two chapters introduced the problem and gave the research goal. While describing the goal of this research, several questions were defined which need to be answered to reach this goal. The first question focusses on the problem related to generating system topologies using network theory. The tool used to do this is the SDS-ATG. This chapter will discuss the tool to show how the current research fits within the whole, and to find an answer to the first research question (RQ1).

3.1. Ship Distributed Systems Automatic Topology Generator (SDS-ATG)

SDS-ATG is a tool that is a combination of an optimization algorithm (NSGA-II, [Deb et al., 2002]) and a tool used to evaluate a function X. A flow diagram of the SDS-ATG tool can be seen in figure 3.1.

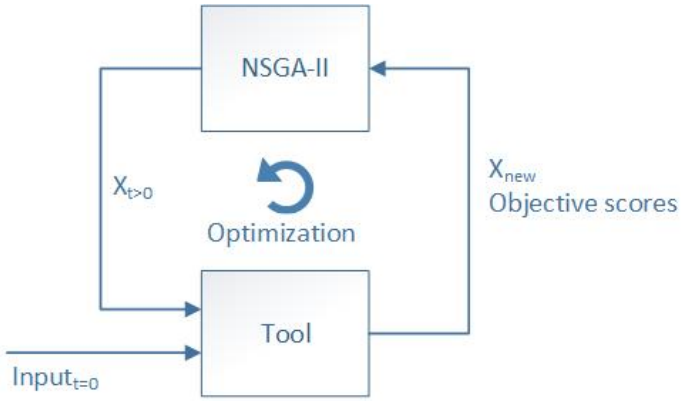


Figure 3.1: Flow diagram of the SDS-ATG tool

The general optimization process can be described in four steps.

1. At the first time step ($t = 0$) an input is needed from the user. This input can be either a starting point identified by the user or a random starting point. Furthermore, the input can be a single topology or a set of topologies.

2. Using the user input at $t = 0$, the tool as seen in figure 3.1 generates the corresponding system topologies. These topologies are then evaluated and given the objective scores, which determine their identity (e.g. size and vulnerability). Besides the identity of the input system topologies, the tool also sends the function X_{new} which describes the system topologies to the optimization algorithm NSGA-II. This function X_{new} is being sent, as the repair function within the tool can alter the input to have it correspond to the boundary conditions of the generated system topologies, thus the updated X (X_{new}) should be communicated with the optimization algorithm.

3. The NSGA-II algorithm indexes the incoming system topologies using the objective scores. Then using all of the generated information, the algorithm creates a new set of system topologies as the function $X_{t>0}$. This new set is then sent to the tool for possible repair and evaluation.

4. With the new input generated from the NSGA-II algorithm being sent to the tool, a loop is formed linking the tool and the algorithm. This loop continues to increase its iterations until a desired outcome or the maximum number of iterations has been reached.

The above steps give a short explanation of optimization process. A more detailed description of the NSGA-II algorithm will be given in section 3.2. A more detailed form of the tool from figure 3.1 can be seen in figure 3.2.

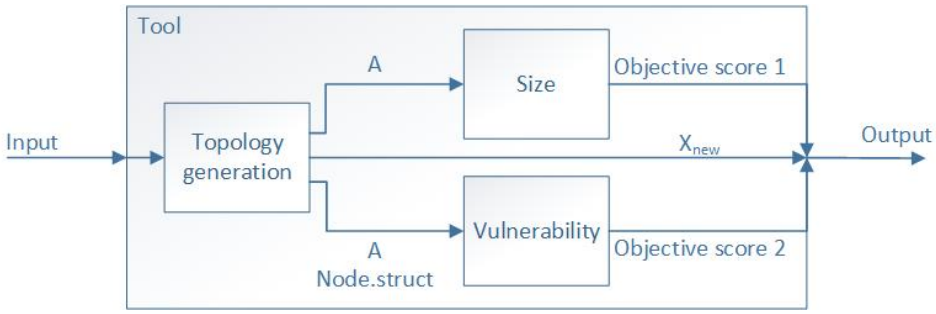


Figure 3.2: Flow diagram of the topology generation and objective evaluation

Figure 3.2 shows in more detail how the objective functions are generated using the input into the tool. The input is, as mentioned above either, a user generated function or a function created by the NSGA-II algorithm. The function that is used for the SDS-ATG tool is a vector X. This vector shows the information on how the topologies should be generated (this is explained in section 3.4.2. With this information, the topology is generated and repaired where needed. The topology generation is discussed in section 3.4, while the repair needed is discussed in section 3.4.3. With the correct topologies generated (the topologies are repaired and thus fulfil the topology constraints), the evaluation for the different objective functions can begin. In this case the objective functions are: the size and the vulnerability of the topology. As mentioned in chapter 1, the size of a topology can be related to the cost of that topology, this is also discussed in section 3.3.1. The vulnerability function evaluates how prone the topology is to failure due to either internal (bad maintenance) or external factors (weapon damage or collision). As mentioned before, the output of the tool in the form of the new X vector, X_{new}, and the two objective scores of the topologies are sent to the NSGA-II algorithm.

3.2. NSGA-II

The optimization algorithm used within the SDS-ATG tool is the NSGA-II or Non-dominated Sorting Genetic algorithm [Deb et al., 2002]. This optimization algorithm is capable of handling multiple objective functions as well as constraints. The capability of this algorithm to handle multiple objective functions to optimize the given topology is used by the SDS-ATG tool.

The idea of the NSGA-II algorithm is to vary the input used to generated the objective functions (in this case two) in such a manner that the solutions of the topologies converge to both objectives. This creates a so-called Pareto optimal front between the two objective functions. On this front optimizing one objective function leads to a decrease in the other objective function. For an example of the pareto front see figure 3.3

The genetic algorithm can be described by four stages that it goes through every iteration. The four stages are; evaluation, selection, crossover and mutation. After these four stages the old population given as an input is adjusted to a new population in the next generation. This new population can then be evaluated after which the process begins anew in the next iteration. A flowchart of the process of a generic genetic algorithm can be seen in figure 3.4.

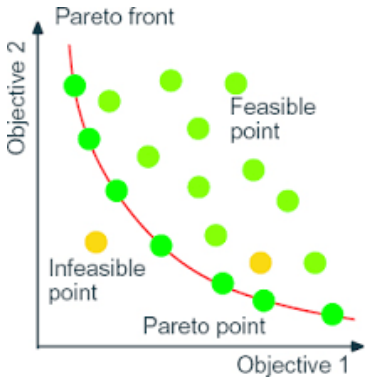


Figure 3.3: Example of a pareto front [Gollub and de Vivie-Riedle, 2009]

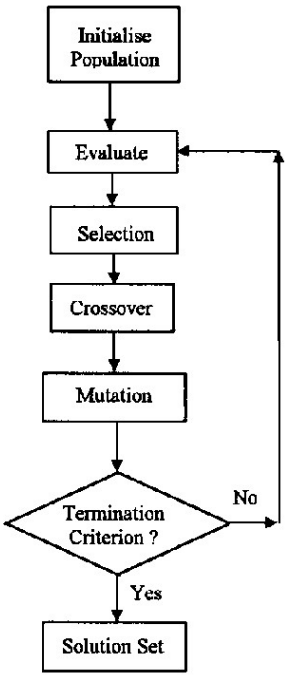


Figure 3.4: Process of a genetic algorithm [Kunjur and Krishnamurty, 1997]

3.3. Objective scores

The objective scores are used by the NSGA-II function to determine how the topologies measure against each other. The objective scores can be literally anything if they describe a property of the topology. However, during this research two types of functions can be defined that will be used to identify the topologies. The types of functions are: the cost of the topology as seen in section 3.3.1 and the vulnerability of the topology as discussed in section 3.3.2.

3.3.1. Cost

The cost of a design is difficult to predict during the preliminary design stage, however with respect to the system design something can be said about the cost between different designs. Especially when the only difference between designs is the amount of connections within the topology. Hence, the cost between different topologies will be related to the size of the topologies. Two functions can be used to determine the size of the topology:

1. The first method is simple function that determines the size of the topology solely based on the number of connections present in the topology. This function should only be used as an indicator during the early stage of the design, as this function has several downsides. The connections are different, there are

connections of the $440V$ network as well as chilled water pipes. These different connections constitute to different costs per connection. Besides the types of connections having different costs, the length of the connections is also unknown in the function and thus the cost corresponding to the length of the connections.

2. The second method uses more detail. This method uses not only the amount of connections within the topology but also some sense of the length of these connections. This method can be used when either the exact position of components is known or when a more global position is known. For instance, if it is known that a vessel will have three different zones; aft, mid and forward. Even with this amount of detail it can help the program to differentiate between solutions that want to lay connections between the furthest components and those that optimized to connections that are nearby. This kind of behaviour will not only help to choose a cost-effective topology, but due to a limited amount of placement information the actual placement will be more logical. In later stages when system positions are known, a more exact estimation of the total length can be made. One of the problems of the first method remains, namely that all connection types are taken equal, even when they are not.

3.3.2. Vulnerability

The second objective score is based on the vulnerability of the system topologies. Since this objective score is the basis for the work performed during this MSc thesis, the different functions to determine this score will be discussed in more detail in chapter 4. This section will deal with the reasoning behind using the vulnerability as an objective score.

To assess a topology on its performance it was decided to use the reliability, see section 1.3.1, of the system topology as it is closely related to the availability of the vessel. To determine the reliability of the topology, it is desired to know how the system reacts to failure of connections or components. The reaction of the system due to failure can be either predicted or estimated, and it is used to determine the vulnerability. A prediction can be made using the an unaltered system topology. This prediction is based on the characteristics found from the network topology and shows how different networks would weigh against each other based on that chosen characteristic. A estimation will be gained from modelling the damage done to the topology and by determining the reaction from this modelled failure. Here, the vulnerability can be the chance of the system not meeting its requirements due to failure. With the reaction to certain types of failure known in the form of the vulnerability, as well as an indication of the chance of said failure to occur, the reliability can be estimated.

In most cases, a vessel that is being judged on its reliability will be subject to failure due to internal factors (maintenance, fire, flooding, etc.). For some of these failures the chance of occurrence can be determined, however this will take extra time during the early design stage if it is possible at all. It would be best to omit the chance of a chosen failure to occur and only look at the performance of a topology when subjected to a certain type of failure. With this omission, the performance indicator of the topology will no longer by the reliability, but the vulnerability.

3.4. Topology generation

Topology generation is one of the key functions of the SDS-ATG tool. De Vos made a function which can construct a system topology using a set amount of input parameters. This function was rewritten to accommodate a more generic approach to the input needed to create a system topology. The rewritten function will be discussed in this section.

3.4.1. Components

When a system topology is to be constructed, knowledge about the system components should already be present. This could be in a lesser variant of detail where a diesel generator could be a component of the system, or in more detail where the diesel generator is the entire system and parts of the diesel generator are the components. In both scenarios, some data about the components is assumed to be known and should be used as the input for the topology creation tool. Currently, the tool uses the following input:

- *Component ID*

  The components ID is a value given by the program to the component. It is used for the component throughout the program and matches the row and column corresponding to the component in the system matrices both index and adjacency.

- *Component name*

  This shows the identity of the component so that it can be recognized when looking at the data afterwards.

- *Network type*

  A component can be part of one or more distribution networks. For instance, a server is a part of the electric (it uses electricity), the chilled water (it requires cooling) and the data network (it processes data). The network type is the name for the networks the component is a part of. This name is also linked to a Network ID.

- *Network capacity*

  The component will interact with the system in a certain manner. This manner of interaction is shown in the *network capacity* of a component within a *network type*. The *network capacity* is given as the amount of energy a component supplies to- or demands from the distribution network of the *network type*. A value for the *network capacity* is necessary, but when no real values are known (such as power in $[W]$ for the $6600V$ distribution network) a simple scale of $-1, 0, 1$ can be used. The Network capacity will be used to split the component into three classes for every Network type.

  1. *Suppliers*

     These supply the given resource (e.g. electricity, cooling water or data) to a distribution system. The capacity of a supplier is always positive.

  2. *HUBs*

     HUBs are the distributors within the distribution system. They do not use any energy nor will they supply energy. However, they will distribute their inputs over multiple outputs. As no energy is (theoretically) lost, their capacity is always zero.

  3. *Users*

     A user uses the energy within a distribution system. Their capacity is always negative. In some cases, a user to one distribution network is a supplier to another. This user that is also a supplier is called a converter, as it converters a used resource into a supplied resource.

- *Location*

  During the concept design phase or any later phase, the location of the component within the ship may be available at various level of detail. This location can be a rough estimate such as: a zonal distribution, a compartment number as intermediate detail or the exact location of the centre of gravity of the component when it is available in a later stage of the design process.

3.4.2. Creating the adjacency matrix

From the components given to the topology generation tool, an index matrix can be constructed. The index matrix shows which connections are theoretically possible for the network. It does this by determining which components are present within each distribution network. In the different distribution networks, it is then established which of the components are the suppliers, HUBs and users. Each of these three has its own possible connections, these connections are stated below.

- When a distribution network has HUBs
  - Suppliers can only supply energy to HUBs
  - HUBs can distribute energy to other HUBs and to users
  - Users can only receive energy from HUBs
- When a distribution network lacks HUBs
  - Suppliers can only supply energy to users
  - Users can only receive energy from suppliers

From network theory, it can be found that every network that has a flow of goods going through it, has a source and a sink node [Hillier and Lieberman, 2010]. From the above it can be concluded that the suppliers are the sources and the users are the sinks. For this reason, it should be logical that it is impossible for a source to supply energy to another source. Also, it is illogical for a user to receive energy from another user within the same layer type.

The above stated possible relations between the three kinds of nodes within the distribution system can be used to construct the index matrix. The tool constructs this matrix by first determining the size of the $[n \times n]$ matrix using the number of total components in the system, n. Using this matrix size, it generates an initial index matrix containing all the indices. This initial matrix is than transformed to the real index matrix by deleting all the impossible connections based on the imbedded system logic explained above. *Example 1* shows this process.

Example 1:
A distribution network contains two suppliers $(S1, S2)$*, two HUBs* $(H1, H2)$ *and two users* $(U1, U2)$*. The initial and real index matrix are shown below. A graphical representation of the fully connected distribution network can be seen in figure 3.5*

$$B_{initial} = \begin{bmatrix} 1 & 7 & 13 & 19 & 25 & 31 \\ 2 & 8 & 14 & 20 & 26 & 32 \\ 3 & 9 & 15 & 21 & 27 & 33 \\ 4 & 10 & 16 & 22 & 28 & 34 \\ 5 & 11 & 17 & 23 & 29 & 35 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{bmatrix} \quad B_{real} = \begin{bmatrix} 0 & 0 & 13 & 19 & 0 & 0 \\ 0 & 0 & 14 & 20 & 0 & 0 \\ 0 & 0 & 0 & 21 & 27 & 33 \\ 0 & 0 & 16 & 0 & 28 & 34 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

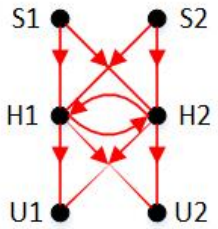


Figure 3.5: Example 1, distribution network

With the index matrix for a fully connected system created the next step is to make an index vector. This index vector has a length based on all the non-zero values in the real index matrix. Each of the places in the index vector will be assigned a one or a zero. This assignment can either happen by a random number generator, when the users wants to create a random topology. Or the assignment can happen by the user or NSGA-II function.

In the index vector a one means that the connection corresponding to the index number of the index matrix exists, while a zero means that no connection is present. Since the index vector entries correspond to the real index matrix the assignment of ones and zeros creates a possible adjacency matrix. This process can also be seen in *example 2*.

Example 2:
Using the real index matrix from example 1, the index vector x_{rel}*, corresponding to this index matrix looks as follows:*

$$x_{rel} = \begin{bmatrix} 13 & 14 & 16 & 19 & 20 & 21 & 27 & 28 & 33 & 34 \end{bmatrix}$$

Using the relation between x_{rel} *and* B_{real} *the initial adjacency matrix can be formed. The vector* x_{rel} *has been filled with zeros and ones and the corresponding initial adjacency matrix,* $A_{initial}$*, is shown. Figure 3.6 shows the distribution network that belongs to the initial adjacency matrix.*

$$x = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A_{initial} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

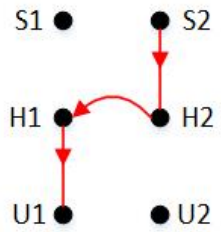


Figure 3.6: Example 2, distribution network

However, in the above figure a network is shown that cannot be correct. The network shows that at least one supplier and one user are not connected. The connection between the HUBs is also a directed connection (a one-way flow), while this connection should be undirected (bidirectional flow). Since the network shown in figure 3.6 is not satisfying, a repair function is needed to make sure that a (randomly or automatically) generated network adheres to the basic system logic.

3.4.3. The repair function

The implemented repair function is meant to repair the faults that could arise when (randomly or automatically) generating a system topology using the index vector. Repair is needed in three areas, two of which are shown by figure 3.6 as this figure has an unconnected suppliers and users, and a directed HUB-HUB connection. The three areas for repair are:

1. To connect every component in the system
2. To make every HUB-HUB connection undirected
3. To create certain hard-choice connections (the user override)

Each of these repair areas will be dealt with in more detail below

Connect every component

The first repair area solves the problems regarding to unconnected components, since an unconnected component is useless and could be left out of the system. However, the components where all stated to be within the system and thus they must be connected. To repair the initial matrix by connecting every component, to following four steps must be taken for each component.

1. Determine which distribution networks the component is a part of
2. Determine the function of the component within these networks (supplier, HUB, user)
3. Based on the function of the component, check if it is connected in the manner described in section 3.4.2 If there is not a single connection, go to step 4.
4. Using the information of the distribution network, randomly [1] generate a connection between the component and one of the other components it can be connected to.

[1] Normally, a randomly generated input for an optimization program is not possible as the input cannot be reproduced. However, the information concerning the final network will overwrite the first introduced index vector and will be sent to the optimization function making it possible to reproduce the network used.

These four steps should be repeated until every component has been examined. *Example 3* will show how these repairs are done for the network of figure 3.6 which is in need of repairs.

Example 3:
The system topology of figure 3.6 has two components that are disconnected, namely: the first supplier ($S1$) *and the second user* ($U2$). *The other components have at least a single in- and output within the distribution network and are thus connected.*

For both components, the connection to the HUB is missing. This problem can be resolved by taking a random number between one and the number of available HUBs for both disconnected components. This random number will tell the program to which HUB the component should be connected. The process is shown below.

$$\begin{bmatrix} S1 & U2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \end{bmatrix}$$

From the above vector if follows that both components are to be connected to H2. This gives the following adjacency matrix. The network belonging to this matrix is shown in figure 3.7

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

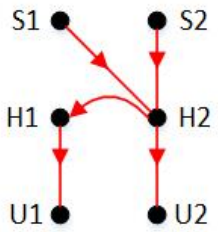


Figure 3.7: Example 3, distribution network

Undirected HUB-HUB connections

The next step of repair is to create undirected HUB-HUB connections within the network. The connections between HUBs should allow free flow of energy, which is false when a HUB-HUB connection is directed.

The repair function determines which of the components in the system topology are HUBs. Next the connections between these HUBs are identified. Knowing which connections are between HUBs, the repair function determines whether the connection is symmetrical in the adjacency matrix. If the connection is not symmetrical both inputs are made one. For instance, if a HUB-HUB connection exists between two components that have the ID numbers I and J, the function checks if $A(I,J) = A(J,I)$. If this is the case, no action is needed. Otherwise, both $A(I,J)$ and $A(J,I)$ need to be taken as one. The effect of this repair function is shown in example 4.

Example 4:
In figure 3.7 an almost complete network is shown; however, the HUB-HUB connection is still directed. Since the ID numbers of the two HUBs are 3 and 4, the following will happen.

$$A(I,J) = A(J,I) \rightarrow A(3,4) = A(4,3)$$

However,

$$0 \neq 1$$

Thus,

$$A(3,4) = A(4,3) = 1$$

This gives the following adjacency matrix and the network as shown in figure 3.8

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

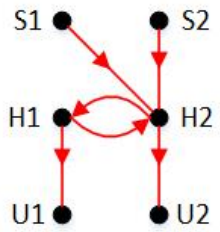


Figure 3.8: Example 4, distribution network

User override

The last repair functions determines certain hardwired (user input) connections. These hard-choice connections can be connections that should always be present. It can also be that these connections are connections that are out of the ordinary, for instance a bypass connection form supplier to user without using a HUB. This last repair function was added to create a possibility to consider situations that fall outside the normal scope of the tool.

3.4.4. Assumptions

Creating a topology using the SDS-ATG tool requires the engineer to define components which are then used to create an adjacency matrix. Finally, a repair function fixes the matrix, so that it corresponds with the constraints given for the topology. To go through this process several assumptions had to be made. These assumptions are discussed in this section as well as the problems that the tool creates and solutions to these problems, answering research question one (RQ1).

1. *Simplification of the system design*

   The first step to model the system design using the SDS-ATG tool is to assume that any component within the system design can be modelled as either a node (component) or an edge (connection). For this assumption to hold, several simplifications have to be made.

   - Every component within the system can be designated as either a supplier/HUB/user of atleast a single distribution network.
   - Suppliers and users are modelled as components with a certain amount of input and output. Everything that happens within these components is disregarded when not specifically identified by the user.
   - HUBs are components that distribute the resources within a distribution network. These components will not absorb resources of the network type that they are HUBs for.
   - The distribution of resources within a distribution layer is accomplished using a HUB. It can be seen that this can be done for a switchboard as it distributes the electricity. However, when taking the piping on board a vessel this HUB cannot be identified as fast. Thus it is assumed that the piping distribution system may be modelled as a HUB.

- The size of components can be neglected as only the approximate location of the component is of importance (when taken into account).
- No switches, valves, etc. are modelled within the connections between between systems. The edge within the topology is only the connection between two components.
- The size of edges can be identified by either network type or edge capacity. The size of edges is found as the length of an edge between two components.

2. *Available information*

   As shown in section 3.4.1 information on the system design should be known. Although the level of detail can still be very low, it assumed that atleast a basic idea of the system design is present (e.g. layer types and number and function of components).

3. *Existence of a connection*

   When a connection is identified within the topology it is present within the vessel. This connection could be disconnected during the actual use of the vessel by a switch, valve, etc. but, it is available. When no connection is shown within the topology network, no connection is present within the system design. Furthermore, when a returning connection is necessary, such as within a pipe system, two are tied together so that no extra edges are needed to explain the other side of the network.

4. *No node loops*

   A component is unable to create a loop from itself to itself using only a single edge. In other words, an edge is always connected to two different nodes, the diagonal of the adjacency matrix is zero.

The above assumptions are the basis for the transformation of system designs to the network topologies as used within this research. However, these assumptions are also the basis of problems that can occur during the generation of network topologies. Two problems are discussed here, together with their respective solutions.

The first problem to address is how to deal with components that cannot clearly be said to be either a supplier or a user. Components that have this criteria are capable of storing energy and discharging it at a later moment (e.g. batteries, capacitors, flywheel, etc.). However, these components will never be both a supplier and a user at the same moment, thus the supplier/HUB/user designation does not work. What could be done to solve this problem is by creating two different scenarios; one in which the components are storing energy (user) and one in which it is discharging energy (supplier).

The second problem is based on the edges identified by the tool. These edges stand for the connection between two components and are given as binaries within the adjacency matrix. However, using the binary assignment a possibility for redundancy of a single edge disappears. For this redundancy during the topology creation, higher values for the edges should exist in either the adjacency matrix or in a separate redundancy matrix. Although this is not a common occurrence, for the critical components an option for redundant edges should be available.

4

Estimating the vulnerability of a system topology

This chapter discusses different methods and metrics that can be used to define the vulnerability of the system topology of a vessel. The chapter has been divided into four sections. The first section gives a short introduction in the terminology used in this chapter. The second section shows a few examples of standardized metrics used in the literature to define the vulnerability of networks. Thirdly the method to determine the performance of the system topology as proposed by de Vos et al. [2017] is discussed. Finally, the developed method to determine the vulnerability is explored.

4.1. Network theory terminology

Since most of the metrics and methods used are based on the network theory, the following terms and abbreviations should be known.

- *Node*, a component as discussed in chapter 3
- *Edge*, a physical connection between two nodes, such as a cable or pipe
- *Network*, combination of nodes and edges (e.g. a distributed system)
- *Adjacency matrix*, matrix that shows which nodes are connected with edges and in which direction
- *Distribution network*, network within the system topology for which every component uses the same resource type: e.g. electricity, chilled water, mechanical translation or data
- *Graph*, visual representation of a network
- *Distance*, number of edges within the shortest path between two nodes

4.2. Network examination metrics

Network examination metrics, are the metrics based on the current network theory. These metrics have different approaches to examine a chosen network, but they are all centred around the topological characteristics of the network. Since the networks that describe a system topology are influenced by system logic, some of the metrics have no merit. This section will explain three metrics, namely:

1. *Characteristic length*
2. *Degree centrality*
3. *Edge betweenness*

After discussing the three metrics, a conclusion can be formed about the usefulness of these metrics in predicting the vulnerability of system topologies.

4.2.1. Characteristic length

The characteristic length or the average shortest path mentioned in several papers [Guohua et al., 2008, Xu et al., 2009] is a metric that uses the number of edges between two nodes to determine the robustness of a network. For a simple (i.e. a network without system logic) undirected network, the formula for the characteristic length, L, can be seen in equation 4.1.

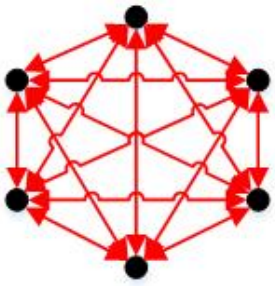
$$L = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} d_{ij} \tag{4.1}$$

The above equation uses the average of the minimum path between two nodes to determine the characteristic length. It does so using the shortest path, d_{ij}, between node i and node j, and the total number of nodes, N, within the network. This shortest path is measured as the number of edges between the two nodes (i.e. distance).

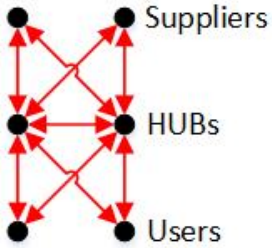
However, the networks showing the system topology cannot be classified as simple undirected networks. The reason for this is that the system topologies are directed networks. When applying equation 4.1 to directed networks, it can be possible that $d_{ij} = \infty$ for a certain i and j, as no possible path between these two networks exists. However, this is not shown in the averaging by all possible node pairs, thus giving an infinite characteristic length. A new formula was proposed [Mao and Zhang, 2013] which is shown in the equation below.

$$L_{new} = \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} d_{ij}}{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} D_{ij}}, \qquad D_{ij} \begin{cases} 1, & d_{ij} \neq 0 \\ 0, & d_{ij} = 0 \end{cases} \tag{4.2}$$

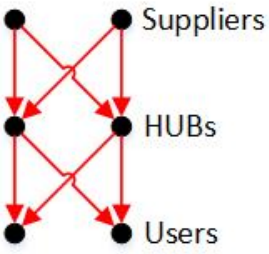
This new characteristic length also uses the sum of the length of the shortest paths, d_{ij}. However, to avoid to possibility of L_{new} going to infinity d_{ij} has been taken as zero if node i = node j or no path exists between node i and node j. Using the newly found sum of d_{ij}, the equation now uses the number of existing node pairs D_{ij} to find the characteristic length. The new characteristic length L_{new} thus uses the average shortest path length of all existing paths. The difference between the original characteristic length and the new one are shown in figure 4.1 and table 4.1. The networks shown in the figure show the development from a fully connected undirected 6 node network 4.1a to a fully connected 6 node distribution network 4.1d.



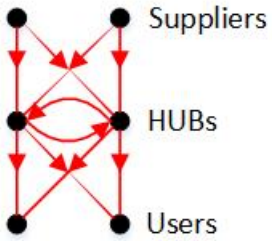
(a) Fully connected undirected 6 node network



(b) Fully connected undirected 6 node distribution network



(c) Fully connected directed 6 node distribution network



(d) Fully connected 6 node distribution network

Figure 4.1: Networks with different characteristic lengths, see table 4.1

Table 4.1: The characteristic lengths of the networks shown in figure 4.1

Network	L	L_{new}
A	1	1
B	$1\frac{2}{5}$	$1\frac{2}{5}$
C	∞	$1\frac{1}{3}$
D	∞	$1\frac{2}{7}$

As mentioned at the beginning of this section, the characteristic length can be a measure of the robustness of a system. This robustness is based on the size of the length L, with $L = 1$ being the most robust network (each node is connected to every other node with only a single edge) and indicating a less robust network with increasing L. For system topologies, this would mean that high L are possible since the distance between the main suppliers and final users can be relatively large. However, the more sparsely a network is connected, the higher the value of L will be. When more connections are added to the network, L is lowered as shortest paths get shortened.

The equation of L_{new} has a disadvantage, it would deem a network separated network with parallel trees most favourable, as it has a good average shortest paths over existing paths ratio. When cross connects are added, the length of the paths grows faster than the number of possible paths thus giving a higher L_{new} for a more robust network. There is however a saturation point to this effect when all of the trees have been connected. At this point adding new edges would lead to a decline in L_{new}. This saturation point is however hard to determine as well as the networks that lie before it. Due to this abnormality in the results, this method cannot be used to predict the vulnerability of system topologies.

4.2.2. Degree centrality

The degree centrality is most likely the simplest of all the centralities and maybe also the oldest. The idea of the degree centrality is to judge a network based on the amount of edges linked to each node. For the directed networks, a distinction can be made between two forms of the degree. These two degrees are:

- *Indegree*, number of incoming edges to a node
- *Outdegree*, number of outgoing edges of a node

The two degrees determine the importance of every node within the network. Each node can be described by its in- and outdegree giving an overview of the entire network. Figure 4.2 shows a network and table 4.2 shows the degree distribution of the nodes within the network.

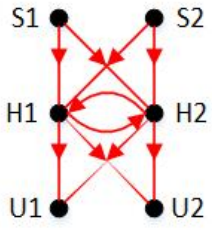


Figure 4.2: Fully connected 6 node distribution network

Table 4.2: Degree distribution of figure 4.2

Node	Degree pair (IN,OUT)
S1	(0,2)
S2	(0,2)
H1	(3,3)
H2	(3,3)
U1	(2,0)
U2	(2,0)

Table 4.2 shows an evenly divided degree distribution, each node in each network layer has the same amount of in- and outdegrees. From this distribution, not much can be said over the nodes themselves besides that the HUBs are the most important nodes due to them having the highest degrees. A change can be seen when a different situation is examined, as seen in figure 4.3 and table 4.3.

This new situation shows a network that is not symmetrical as it was in the first situation. The most important node is now H2, since it has the highest total degree. Figure 4.3 also shows this conclusion, when H2 is damaged or fails both users are no longer reached by the suppliers.

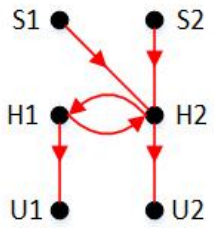


Figure 4.3: Connected 6 node distribution network

Table 4.3: Degree distribution of figure 4.3

Node	Degree pair (IN,OUT)
S1	(0,1)
S2	(0,1)
H1	(1,2)
H2	(3,2)
U1	(1,0)
U2	(1,0)

Although this method could point out the most important nodes within a network, is has some problems that should be considered.

- The highest degree of a node does not necessarily mean that the node is the most important node of a network. The second example shows a network where the highest degree describes the most important node. However, the first situation shows an example where the loss of one of the highest degree nodes, the HUBs, will not lead to failure. But, when a supplier or user is hit in the first situation a loss of capacity in the system occurs, which could lead to a loss of functionality.
- The metric does not consider the different distribution networks within the system topology when assessing the entire network. This could be solved by running the metric per distribution layer, but then information of the different suppliers and users will be lost.

Finally, the degree centrality describes the importance of every node, but it does not give a clear description of the vulnerability. Thus, it is pointless as an objective function for the optimization algorithm.

4.2.3. Edge betweenness

The edge betweenness [Boccaletti et al., 2007] is used to identify the importance of edges in a network. The fundamental idea behind this metric is to look how many times each edge is used in a shortest path between two nodes. This can be represented by the following formula:

$$b_l = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \frac{n_{ij}(l)}{n_{ij}} \tag{4.3}$$

This formula shows that the line betweenness b_l is equal to summation of, the number of shortest paths $n_{ij}(l)$ between node i and node j through edge l divided by the total number of shortest paths n_{ij} between node i and node j.

Basically, if a single edge is often part of the shortest paths between all node pairs in the network its importance becomes higher. For example, this is the case when there are two network clusters which are separated with a single edge. In this case, that single edge is all that connects two large parts of the network. This connecting edge will have a much higher edge betweenness than the other edges in the network.

However, problems are still present with this metric that make it hard to implement it into MATLAB. These problems are:

- The methods used in MATLAB are deterministic, this means that no randomness is involved. In a deterministic system, each set of input parameters will always have the same solution. With such a system, it will be impossible to generate n_{ij} and $n_{ij}(l)$.

  To generate the total number of shortest paths within the system it is thus necessary to create a function that determines all possible input parameters for a certain network. However, the amount of possible shortest paths in a network can grow at a terrifying rate. This holds true for larger networks, as the increasing number of nodes also increases the amount of paths needed to be found.

- Due to the system logic in the generated system topologies, a shortest path is not always a good example to find important edges. This can be relevant when a node is supplied by two different distribution networks. If one of these distribution networks is supplied by the other, for instance; electric and chilled water network. The shortest path will always circumvent one of the distribution systems. This may give a misinterpretation of the importance of edges as some edges are forgotten. This could be solved by defining an edge betweenness for every distribution network, but this creates a more complex problem.

The problems related to the edge betweenness can partly be solved by defining the average edge betweenness $b(G)$ of graph G, as seen in equation 4.4. Basically, it is the sum of all the edge betweennesses divided by the total number of edges $|E|$.

$$b(G) = \frac{1}{|E|} \sum_{l=1}^{E} b_l \tag{4.4}$$

As described by Boccaletti et al., the above average edge betweenness can be rewritten to the formula shown in equation 4.9. This is done using the following steps.

1. The line betweenness b_l from equation 4.3 can be filled into equation 4.4.

$$b(G) = \frac{1}{|E|} \sum_{l=1}^{E} \left(\sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \frac{n_{ij}(l)}{n_{ij}} \right) \tag{4.5}$$

2. The above equation can be rewritten to the form seen below.

$$b(G) = \frac{1}{|E|} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \frac{1}{n_{ij}} \left(\sum_{l=1}^{E} n_{ij}(l) \right) \tag{4.6}$$

3. The number of shortest paths between two nodes through an edge, $n_{ij}(l)$ can be described as:

$$n_{ij}(l) = \sum_{g=1}^{P_{ij}} X_g(l) \tag{4.7}$$

   With P_{ij} being all of the shortest paths joining node i and node j. If l is on the shortest path g, then $X_g(l) = 1$, else it is 0.

4. Filling in $n_{ij}(l)$ will cause the average edge betweenness to take the following form:

$$b(G) = \frac{1}{|E|} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \frac{1}{n_{ij}} \left(\sum_{g=1}^{P_{ij}} \sum_{l=1}^{E} X_g(l) \right) \tag{4.8}$$

   With the distance d_{ij} the formula can be rewritten to its final form as seen in equation 4.9.

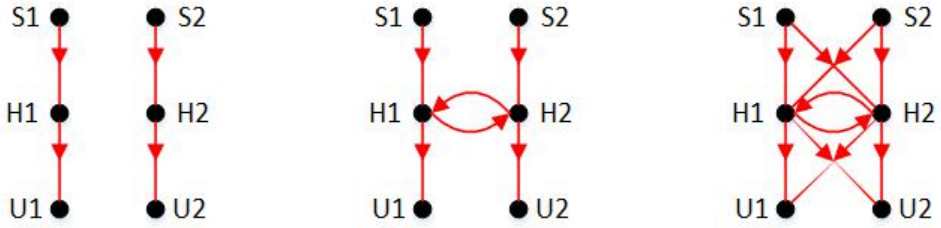
$$b(G) = \frac{1}{|E|} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \frac{1}{n_{ij}} \left(\sum_{g}^{P_{ij}} d_{ij} \right) \tag{4.9}$$

This new formulation of the average edge betweenness, found by Boccaletti et al., uses the sum of the shortest paths d_{ij} between node i and node j. This sum is then divided by the number of shortest paths n_{ij} between i and j. In other words, dividing the sum of shortest paths by the number of shortest paths to get the shortest path d_{ij}. Then the summation of all the shortest paths in the network is taken divided by the total number of edges $|E|$. It is thus that equation 4.9 can be rewritten to the formula shown in equation 4.10.

$$b(G) = \frac{1}{|E|} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} d_{ij} \tag{4.10}$$

Since equation 4.10 only uses the length of the shortest path between node i and node j, the deterministic methods of MATLAB can be used to find a solution. The time needed to come to this solution is also close to none as the adjacency matrix can easily be used to determine both $|E|$ and d_{ij}.

Finally, there are still problems with the average edge betweenness. These problems exist because of the system logic of the system topologies. Take for example the three networks shown in figure 4.4. All the networks shown below have a different average betweenness value. It can also be estimated that network



(a) Fully developed disconnected network $b(G) = 2$

(b) Sparsely connected network $b(G) = 4$

(c) Fully developed network $b(G) = 1.8$

Figure 4.4: Three examples of a 6 node distribution network and their edge betweenness $b(G)$

4.4a is the most prone to losing capacity after a single point of damage and failure. Network 4.4b is sturdier, because of a HUB-HUB connection. And network 4.4c is the least vulnerable, because it is fully connected.

The problem lies in the existence of the networks as shown in figure 4.4a. This network has an average edge betweenness value close to that of a fully connected network, without being as robust as these networks (a single failure leads to the failure of atleast one user). It is thus because of the disconnected networks that the edge betweenness metric is not capable to be used as a viable way to predict the vulnerability of a system topology, unless another constraint is added. This constraint will demand that every HUB should be able to reach every HUB of the same layer. But, such a restriction would rule out many topological possibilities and is thus chosen not to be added.

4.2.4. Conclusion

The metrics discussed in this section are used to examine networks based on their topological characteristics. These metrics have been considered as they can give a fast prediction of the vulnerability of the system topologies. The first metric to fail is the degree centrality as it cannot be used as an objective function within the SDS-ATG tool. The other two discussed metrics, show to be useful for ordinary networks as found within network theory. However, when the system logic was introduced a problem occurs in the metrics.

The problem is that parallel networks or fully developed disconnected networks as shown in figure 4.4a. These networks are predicted to be better than networks with cross connects such as the network shown in figure 4.4b even though its chances to fail are higher.

From the metrics, it can be shown that improvements can be made and lessons can be learned. These are as follows:

- A generalization of the system topology networks cannot be used, as the system logic leads to different outcomes. Shown by the example given above.
- Deterministic methods should be used in the creation of the vulnerability method, as the applicability of these methods are better suited for MATLAB as used within this research.
- Every node should be individually evaluated to find its importance and the effect of the node on the whole of the network.

4.3. Maximum HUB flow method

The method discussed here was proposed by the maker of SDS-ATG as a metric to identify the robustness of a topology considering the involved system logic. The method is based on the flow through the system or to be more precise: the maximum flow the HUBs are capable to offer. This section is divided into three parts to get a better understanding of this method. The first part focusses on the idea behind the method. The second part shows how the method is applied to the system topologies. Finally, the advantages and disadvantages of the method are discussed.

4.3.1. Why maximum flow?

The idea behind this method is that reconfigurability and therefore robustness of a system topology is highly dependent on HUB layer topology. It is thus assumed that the maximum flow that can be generated in the

HUB layer is a good measure for the vulnerability of the system. This assumption is founded on experience and knowledge of system design on board of vessels. The following assumption had to be made to find the reason why this maximum flow method works.

1. Star networks are not desired, which is the type of network in which a node is connected to all others.
2. HUBs are the main component to determine the reconfigurability of the system.
3. Components that are the users and the suppliers in a distribution network are mostly coupled to only one HUB.
4. Suppliers and users are mostly standing close to their distribution HUB.
5. Components with the same task are never close enough to be hit at the same time.
6. Analysing a non-damaged system topology can give a good prediction of the robustness of the system topology.
7. The metric should be usable for both directed and undirected networks.

The above assumptions show the desire of a non-star network. This desire is fulfilled by connecting the HUBs while keeping only a single connection from suppliers/users to HUBs. Furthermore, the assumption of a prediction being able to acquire the robustness of an network results in the usage of this metric. With the second assumption, it is shown that the HUBs should be used for the prediction. Due to the reconfigurability of the network through its HUBs, it can be seen that this method should be used.

4.3.2. Application of the method

The method is applied using MATLAB and the SDS-ATG tool. The application is split in three distinct parts; input, method, output. These parts will be discussed here.

Input

SDS-ATG produces the adjacency matrices of the system topology which can be used in several ways to get results. This method uses the following input:

- *Adjacency matrix*
- *Node information*

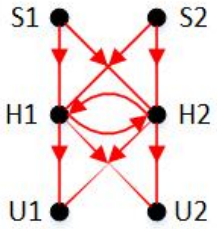
Method

The above shown inputs are provided to the system. The node information, indicates which of the nodes are suppliers or users. These nodes are then disconnected from the rest of the network for the rest of the evaluation, as the maxflow function is solely interested in the HUB-layer. In the adjacency matrix, this is equal to creating a zero row and a zero column for these nodes. To show this, an example network is used. This example network is a 6-node fully connected network with two users, two hubs and two suppliers shown below together with its adjacency matrix, A, in figure 4.5a. Figure 4.5b shows the deleted suppliers/users and the matrix $A_{maxflow}$ shows the corresponding adjacency matrix.

When only the HUB connections remain the maxflow-function of MATLAB can be used. This function determines the maximum flow from a node i to a node j. This maximum flow consists of all possibilities to generate a flow between two points. When an unweighted network is given, such as is the case in this study, the weight of each connection is taken as one. The flow between two points is one for the network given in the above figure, as only one connection is available. However, if figure 4.6 is considered, the maximum flow is three, as three routes from every node to every other node are possible.

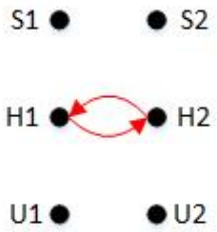
The maximum flow is generated per HUB-layer and a summation is taken of this flow. However, due to the size difference between layers a normalization must happen to generalize the flow within each layer. This creates a situation where every HUB-layer is taken equal to the others. To normalize the maximum flow, it is divided by the total number of available HUBs $N_{HUBs-layer}$ in the layer minus one, as shown in the equation below.

$$Maxflow_{HUB,normalized} = \frac{Maxflow_{HUB}}{N_{HUBs-layer} - 1} \tag{4.11}$$



$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Fully connected network



$$A_{maxflow} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Maxflow network

Figure 4.5: Example of a maxflow network

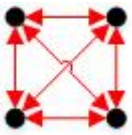


Figure 4.6: Fully connected 4-node network

This normalized flow can be summed to get the total flow in the HUB-layer. This flow per layer must be normalized once more by the number of nodes pairs that are available in the layer. So, if $N_{HUBs-layer}$ is taken as the number of Hubs in a layer, the maxflow per layer is given below.

$$Maxflow_{HUB-layer,normalized} = \frac{\sum Maxflow_{HUB,normalized}}{N_{HUBs-layer}(N_{HUBs-layer}-1)} \tag{4.12}$$

This HUB-layer maxflow can now be summed to create the value used to evaluate the different system topologies.

Output

As shown in the method there is only one value as an output of this function. This value is equal to the summed maxflow of each HUB-layer. The output takes on a different value for every network processed as it is mostly dependent on the number of HUB-layers present in the system topology.

4.3.3. Assessment of the method

This method uses an approach to determine the vulnerability or robustness of the network that is based on knowledge of system design and experience. This method has great value, but also some problems. These advantages and disadvantages will be discussed in this section.

Advantages:

- The method considers some basic logic of system design.
- Due to the normalization process, variations in HUB-layer size can be neglected. This creates a situation in which each distribution network is equal (in size).

- The method is computationally fast.
- The method can be implemented into a function with relative ease, hence it can easily be scaled-up for use on larger system topologies with many HUB-layers.

Disadvantages

- As maxflow only considers the HUB-layers, it does not give a clear description of how the entire system would function when components or/and edges fail. For instance, the functionality of the system can be lost due to a failure of the only user that provided the functionality.
- Each distribution network is taken equal to its peers. However, not every distribution network has the same importance. A larger network could be less significant than a smaller one. Or a certain type of network is more important than another (e.g. a data network controlling heavy equipment against a low voltage network supplying a coffee maker).
- The hard-wired or user override connections are necessary in order to give suppliers/users more than one connection to HUB layer.
- The method shows a characteristic of the topology in the same manner as the metrics of section 4.2, but it includes system logic.

The advantages and disadvantages show that, although this method can give a good indication of how a network could behave, a clear assessment of the actual vulnerability is still missing as actual damage has not been simulated. The advantages show that system knowledge has been put into the method to filter many different system topologies. However, the disadvantages also show that this method has some problems that cannot be overlooked. Based on the maxflow method, a new vulnerability method has been developed that has been made to meet the requirements of a method set by the disadvantages of this method. This new method will be elaborated in the next section.

4.4. Vulnerability assessment function

Based on the methods and metrics discussed before, a new method was created to assess system topologies. This method defines the vulnerability of a network by observing the consequences of failure (e.g., the loss of one or more components or connections due to internal or external factors) within a system topology. Besides modelling the damage and showing if the system would meet its requirements, the new function also considers the system logic and the problems encountered in the previously discussed methods and metrics, such as:

- Preference to create fully connected HUB-layers before adding edges to suppliers or users (maxflow)
- Incapability to handle the system logic of system topologies (metrics)

The vulnerability assessment function is used by the SDS-ATG tool to determine the vulnerability objective score of the system topologies. Figure 4.7 shows how this function fits into the SDS-ATG tool. The integrated approach uses either the user input or NSGA-II input to create a system topology. This system topology is the first step to determine the vulnerability of this topology.

From the topology creation function, the data concerning all the nodes and the adjacency matrix are taken and passed on to two smaller functions: generation of HIT-matrices and defining prime users (components for observation chosen by the engineer). The first function generates the HIT-matrices (matrices that show failure or damage of nodes and/or edges) used by the vulnerability tools. These HIT-matrices will be discussed in detail in section 4.4.1. The HIT-matrices get created based on the user input on the number of hits (simultaneous failures or damages to the network) needed for observation and the sample size. The first variable, the number of hits is free to be chosen by the user. The sample size however, fulfils an important part in generating realistic values for the objective score. How to determine the sample size and the influence of the sample size will be elaborated in section 4.4.2.

The second part, defining the prime users, defines the components that are under observation by the vulnerability tool (these components are chosen by the user and do not have to be the vital users of the system). The vulnerability formulation used depends on the availability of these prime users (components) within the system design. For example, in order for a naval vessel to be capable of firing and tracking a missile,

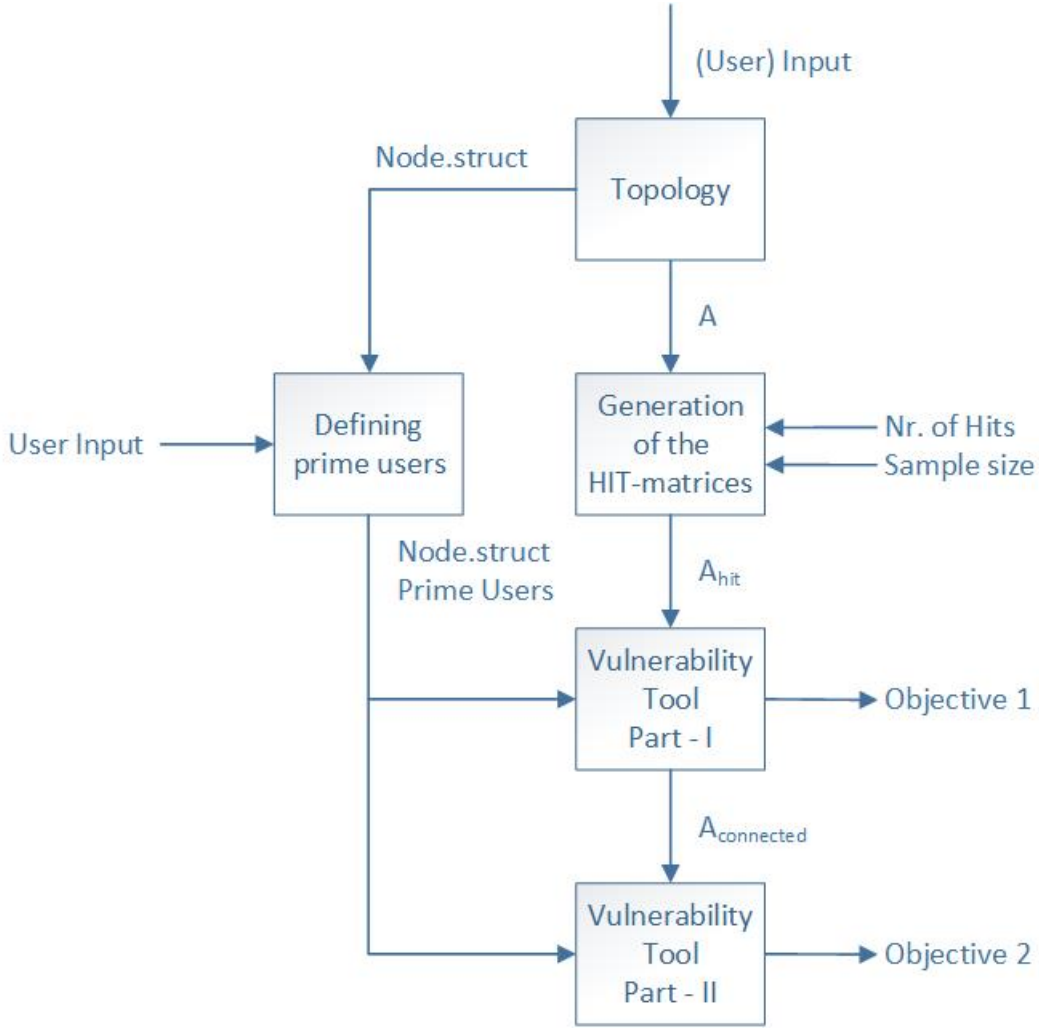


Figure 4.7: Flow diagram of the vulnerability method

the prime users which must be available are: the sensor (radar) and the missile launcher. These prime users are chosen by the user and passed on together with the rest of the node information.

When the first two parts of the method have done their work, the vulnerability tool can use their outcomes to create the objective values needed. The vulnerability tool is a two-layered tool. It exists out of two layers of which the first determines the vulnerability of the topology based on the availability of a path from the suppliers to the components under investigation. This first part defines whether a network is still connected in the appropriate manner. When a topology is still connected, it can be send to the second layer. This second part of the tool determines whether the connected matrix is still able to fulfil the energy requirements of the components under investigation. Part-I and Part-II of the vulnerability tools will be further elaborated in section 4.4.3 and section 4.4.4.

4.4.1. Generation of HIT-matrices

The vulnerability tool uses damaged network topologies to determine the objective score of the non-damaged topology. These damaged topologies are adjacency matrices based on the original topology used, but with several edges and/or nodes removed. These removed edges and/or nodes simulate the damage that has occurred within the network. These damages may have occurred due to internal influences (e.g. maintenance or failure of equipment) or due to external influences (a weapon impact) to the network. This section explains the process of creating the damaged topologies or HIT[1]-matrices.

[1] The term "hit" does not imply that damage is always due to a weapon impact. As explained in the text. Hence, in this thesis "hit" refers to damage due to external or internal influences.

Determining the hit location

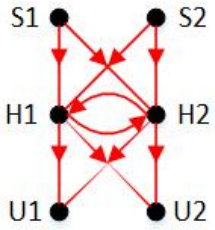
As explained, the HIT-matrices are damaged matrices based on a single topology that simulate the damage taken by the vessel. To simulate the hits, a random function is used since (almost) no geographical information of the ship is assumed to be present. The assumption of no geographical information is based on the current stage of design. A system topology is namely a layout that only shows how certain components in the system are connected. The small amount of information on the location of these components is in most cases not enough to determine if there is a bigger chance to hit a certain system (e.g. a hit on a vessel will have a non random hit pattern on the nodes/edges, given their location on the vessel). This gave the option of a completely random approach to hits to make sure that every possible scenario is considered. Thus, these random hits have an uniform chance to hit every node or edge within the system.

How to process a hit

When a hit occurs, there are two kind of hits. The differentiation between the two types of hits are based on what they represent. The two types of hits are:

- *Edge hit,*
  The edge (connection) is hit and fails. This means that a connection between two nodes, as shown in the adjacency matrix, is removed. For this connection both sides need to be removed, and thus two edges need to be put to zero in the adjacency matrix. If one of the edges is already zero, it will naturally stay in that state.
- *Node hit,*
  This hit terminates a node (component) in the network. The state of a node can be described by the state of the edges connected to this node. If a node is hit, no flow will go to the node or from the node. This can be interpreted as all edges connected to the node are hit. In other words, to create a situation in which a node is hit, it is necessary to disconnect that node from the entire network.

The two hit cases spoken of above are shown below for a fully connected 6-node network, shown in figure 4.8a. The normal adjacency matrix is shown in figure 4.8b with both the edge and node damage profiles. First the edge hit is shown (in red), as the HUB-HUB connection is hit, see figure 4.8c. Secondly the node hit (in blue) is shown with a hit of node H1 as shown in figure 4.8d.



(a) Fully connected 6-node network

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Adjacency matrix of 4.8a with the damage locations of the edge and node hit shown

$$A_{edge\ hit} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Adjacency matrix with an edge hit

$$A_{node\ hit} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(d) Adjacency matrix with a node hit

Figure 4.8: Different hit locations and their effects

Number of hits

The user can give any number of hits that need to be processed by the function to create HIT-matrices. When multiple hits (damage locations) are taken by the topology, the steps of first identifying the hit location and

then the termination of the edge/node is repeated for the total number of hits. Since the total amount of edges and nodes in the network is used, the chance of a hit on a certain location is equal to its peers. There is no chance that a single edge/node is hit more than once, as this would lead to a situation with less hits. The final matrix with the number of hits specified will be used to determine the vulnerability.

4.4.2. Sample size

To generate the HIT-matrices, the sample size is needed to determine the number of HIT-matrices to be generated to create an acceptable answer for the vulnerability estimated by the tool, while still keeping the time necessary to calculate the values to minimum. The entire reasoning behind the sample size can be seen in appendix A. The used formulas are rewritten from the formula found at Stackexchange [2017].

The sample size generation uses a simple formula as shown by equation 4.13. This formula is based on every outcome of a HIT-matrix to have two options either the network is working or it is not. The chance of either of these options to be chosen is shown by p. In this case $p = 0,5$, since the outcome of the assessment is unknown and that is shown by the 50% chance as no side is favoured. The value of Z is found from the normal distribution and correlates to the accuracy demanded of the outcome. The value of c determines how precise the outcome must be, thus the percentage of outcomes that are within the range describe by the accuracy.

$$ss = \frac{Z^2 \times p \times (1-p)}{c^2} \tag{4.13}$$

With the sample size known, it should be corrected for the population of the sample. This should be done due to the sample size being able to grow larger than the population of the sample for smaller populations. This correction formula is shown below in equation 4.14.

$$ss_{new} = \frac{ss}{1 + \frac{ss-1}{pop}} \tag{4.14}$$

With the sample size known it is now possible to calculate the vulnerabilities using the assessment method.

4.4.3. Vulnerability function part I - Connectivity

The first part of the vulnerability function focusses on the connections within the network topology. To explain how this piece of the function works this section is separated into several parts. Firstly, the input needed is discussed. From the input the inner workings of the function can be explained which are followed by detailed examples on what is and what is not possible using this function. Finally, the output that can be generated by the function will be elaborated.

Input

Figure 4.7 shows which inputs are needed for this part of the function to work. The inputs needed by this first part of the function can be seen below.

- *The distribution networks and the nodes that belong to them,*
  The information on the distribution networks helps in splitting the entire network into smaller more easily solvable networks (one for each type of distribution network). These smaller networks make it possible to use a shortest path algorithm within the function. This will be further elaborated during the explanation of the tool itself.

- *Possible connections,*
  Using only the distribution networks, it is possible to separate the entire network topology into smaller sub-problems. However, the information of all the nodes and their possible connections can give a good estimation if a node is connected to a next layer. These possible connections give the ability to determine which interconnections between the distribution networks are necessary (for nodes that are included into several distribution networks) and if they are available.

- *The (HIT) adjacency matrix,*
  Although the first two inputs can show a lot of information on how the connections should run, the adjacency matrix shows which connections are present.

- *Nodes for investigation (prime users),*
  The information on which nodes should be investigated is the final input into the function (e.g. the tool will evaluate if these nodes are still connected in a damaged situation).

Determining if a node is connected

The idea behind this method is that the vulnerability of a system topology can be determined by deciding whether the system is still fully connected. The connectivity of the system can give a good indication into the vulnerability, as a network that is more likely to be disconnected after failure(s) should not be chosen. Besides giving a good indication, the outcome can be used to isolate the topologies that are still connected after receiving damage for further vulnerability analysis. The Part-II of the method as shown in section 4.4.4 is based on these connected topologies.

The method is shown in the flow diagram of figure 4.9. This flow diagram shows the starting point to be the list of nodes that is to be investigated (i.e. the prime users). This list is used to determine if the investigation is finished by checking if any nodes are left to be investigated. To start the analysis process, a node is taken from the list of prime users (prime-node). When the prime-node has been examined, a node will be taken from the list of nodes to be examined.

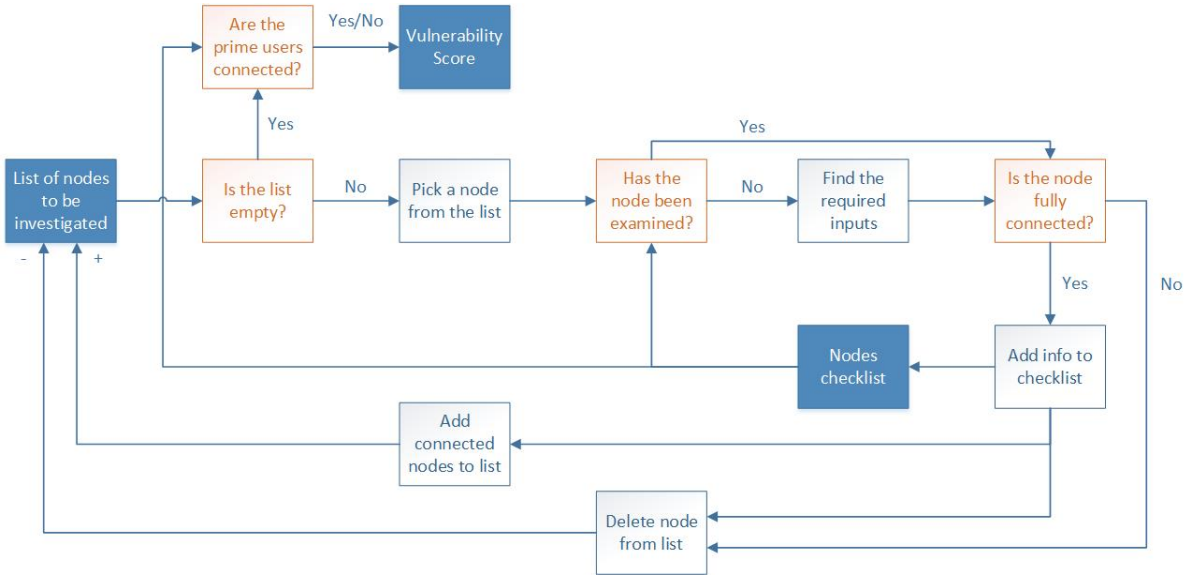


Figure 4.9: Flow diagram part I - Connectivity

The first step is to cross-reference the node taken against the nodes check list, which is filled with the information from already examined nodes. This check list shows if a node has already been investigated and thus if it can be ignored. When a node has not been investigated yet, it is necessary to determine the node inputs. The inputs are determined per distribution network, after which it is investigated whether the node is connected to any suppliers of the given input within the given distribution network. That is, it can be supplied with its required type of resource. For example, a radar needs chilled water and electricity, thus it is investigated if it is supplied with these resources. From this information, it can be concluded if the node is fully connected (e.g. it has connections to its required resource types) or not. This information is then stored in the check list after which the node is deleted from the investigatory list. When the node is fully connected, the nodes connected to it are added to the investigatory list.

The whole process of finding the connectivity of the network is done from bottom to top, with the users at the bottom of each layer and the suppliers at the top. From each node branches are created to its suppliers if it is connected. Due to these branches this method resembles the search tree method of the basic network theory. However, due to the reverse nature of this analysis and the including of system logic (role of users, suppliers and distribution networks) this method is more complicated.

The above method to find the connectivity of a node has its limitations. This method cannot handle a loop between two components. When the two components are dependent on each other, this system of checking the components will let these two components reference to each other in an infinite loop. To ensure the method not getting stuck due to such a loop, a clear hierarchy should be established in the system topology. This hierarchy should contain a clear beginning and end of the topology, thus the engineer is required to determine were to begin its evaluation.

Examples of the connectivity analysis

To elaborate on the process used to find the connections between nodes, a few examples will be shown here. The four examples are based on figure 4.10.

1. Fully connected network, see figure 4.10
2. The first chilled water pump has been damaged
3. The first transformer is damaged
4. The HUB-HUB connection is lost

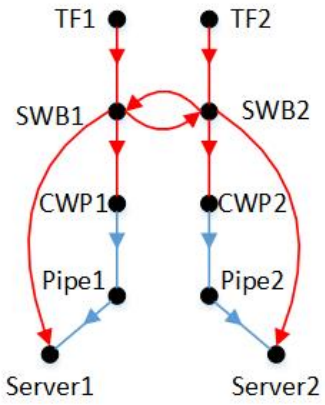


Figure 4.10: Example network for part I - Connectivity, fully connected and undamaged

The network shown above will be used for the examples. The network consists of two different distribution networks; the $440V$ electric network (red) and the $5°C$ chilled water network (blue). The $440V$ network is being supplied by two transformers ($TF1$ and $TF2$) linked to their own switchboards ($SWB1$ and $SWB2$). These switchboards are capable to distribute the received energy to the two attached servers ($Server1$ and $Server2$) and chilled water pumps ($CWP1$ and $CWP2$). The chilled water network consists of two chilled water pumps connected to the two servers by their respective pipes ($Pipe1$ and $Pipe2$), although the pipe HUBs could be taken as edges in this case, they are still given as HUBs due to possibility of a connection between the two pipes. For all examples, the two servers are taken as the nodes under investigation.

Example 1:
The fully connected network as shown in figure 4.10 will be investigated by looking at the two servers. The investigation will begin at server 1. The steps taken by the program to solve this problem are shown in table 4.4. When a node has been investigated, it will be shown in green if it appears again, thus skipping it in the next steps.

At this point every node is investigated, but it is still unclear if the network is connected. To find the connected network, it is necessary to define if users are connected to their suppliers. Since the two transformers are both connected, this part begins in the electric network. The final steps needed to solve this example are shown below, all referenced steps are from table 4.4.

- *Electric network:*
  1. *Step 4 and 6 show that the switchboards are connected to the transformers*
  2. *Step 3 and 10 show a connection between the chilled water pumps and the switchboards*
  3. *Step 1 and 8 show a connection between the servers and the switchboards*

     The electric network is connected
- *Chilled water network*
  1. *The chilled water pumps are supplied by the electric network*

2. *Step 2 and 9 show that the pipes are connected to the pumps*
3. *Step 1 and 8 show that the servers are connected*

   The chilled water network is connected

With these final steps, it is shown that the servers are fully connected.

Table 4.4: Node investigation, fully connected example

Step	Investigated node	Number of connected networks	Connected nodes	Old investigation list	Connected?	New investigation list
1	Server 1	2	Pipe 1 SWB 1	Server 1 Server 2	Unknown	Pipe 1 SWB 1 Server 2
2	Pipe 1	1	CWP 1	Pipe 1 SWB 1 Server 2	Unknown	CWP 1 SWB 1 Server 2
3	CWP 1	1	SWB 1	CWP 1 SWB 1 Server 2	Unknown	SWB 1 Server 2
4	SWB 1	1	TF 1 SWB 2	SWB 1 Server 2	Unknown	TF 1 SWB 2 Server 2
5	TF 1	0		TF 1 SWB 2 Server 2	Yes	SWB 2 Server 2
6	SWB 2	1	TF 2 SWB 1	SWB 2 Server 2	Unknown	TF 2 SWB 1 Server 2
7	TF 2	0		TF 2 SWB 1 Server 2	Yes	SWB 1 Server 2
8	Server 2	2	Pipe 2 SWB 2	SWB 1 Server 2	Unknown	SWB 1 Pipe 2 SWB 2
9	Pipe 2	1	CWP 2	SWB 1 Pipe 2 SWB 2	Unknown	CWP 2 SWB 1 SWB 2
10	CWP 2	1	SWB 2	CWP 2 SWB 1 SWB 2	Unknown	SWB 1 SWB 2

Example 2:
The network from figure 4.10 has suffered damage. Due to the damage the first chilled water pump has stopped working. This new situation can be seen in figure 4.11. Table 4.5 shows the node investigation and the connectivity steps are shown after.

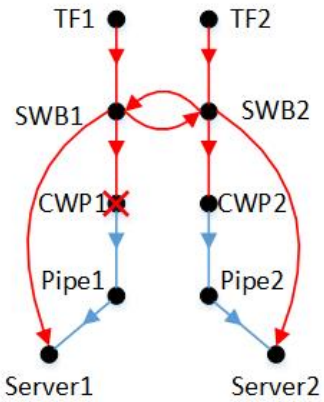


Figure 4.11: Example network of figure 4.10 with the CWP1 hit

Table 4.5: Node investigation, CWP1 hit example

Step	Investigated node	Number of connected networks	Connected nodes	Old investigation list	Connected?	New investigation list
1	Server 1	2	Pipe 1 SWB 1	Server 1 Server 2	Unknown	Pipe 1 SWB 1 Server 2
2	Pipe 1	1		Pipe 1 SWB 1 Server 2	No	SWB 1 Server 2
3	SWB 1	1	TF 1 SWB 2	SWB 1 Server 2	Unknown	TF 1 SWB 2 Server 2
4	TF 1	0		TF 1 SWB 2 Server 2	Yes	SWB 2 Server 2
5	SWB 2	1	TF 2 SWB 1	SWB 2 Server 2	Unknown	TF 2 SWB 1 Server 2
6	TF 2	0		TF 2 SWB 1 Server 2	Yes	SWB 1 Server 2
7	Server 2	2	Pipe 2 SWB 2	SWB 1 Server 2	Unknown	SWB 1 Pipe 2 SWB 2
8	Pipe 2	1	CWP 2	SWB 1 Pipe 2 SWB 2	Unknown	CWP 2 SWB 1 SWB 2
10	CWP 2	1	SWB 2	CWP 2 SWB 1 SWB 2	Unknown	SWB 1 SWB 2

Network connectivity check:

- *Electric network*
  1. *Step 3 shows that SWB1 is connected to TF1*
  2. *Step 5 shows that SWB2 is connected to TF2*
  3. *Step 1 shows that server 1 is connected to SWB1*
  4. *Step 7 shows a connected server 2 to SWB2*
  5. *Step 10 shows CWP2 connected to SWB2*

     Not all the systems are connected, but at this point the program does not recognize this disconnection as it has not considered CWP1.

- *Chilled water network*
  1. *Step 1 shows that server 1 is connected to pipe 1*
  2. *Step 2 shows that pipe 1 is not connected, thus server 1 is not connected*
  3. *Step 7 shows server 2 connected to pipe 2*
  4. *Step 8 shows pipe 2 connected to CWP2*
  5. *This network shows that one server is connected, while one server is not.*

Example 3:
The third situation uses a damaged transformer as seen in figure 4.12. Table 4.6 shows the node investigation followed by the connectivity check.

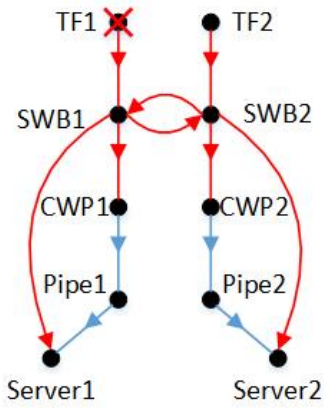


Figure 4.12: Example network of figure 4.10 with TF1 hit

Table 4.6: Node investigation, TF1 hit example

Step	Investigated node	Number of connected networks	Connected nodes	Old investigation list	Connected?	New investigation list
1	Server 1	2	Pipe 1 SWB 1	Server 1 Server 2	Unknown	Pipe 1 SWB 1 Server 2
2	Pipe 1	1	CWP 1	Pipe 1 SWB 1 Server 2	Unknown	CWP 1 SWB 1 Server 2
3	CWP 1	1	SWB 1	CWP 1 SWB 1 Server 2	Unknown	SWB 1 Server 2
4	SWB 1	1	SWB 2	SWB 1 Server 2	Unknown	SWB 2 Server 2
5	SWB 2	1	TF 2 SWB 1	SWB 2 Server 2	Unknown	TF 2 SWB 1 Server 2
6	TF 2	0		TF 2 SWB 1 Server 2	Yes	SWB 1 Server 2
7	Server 2	2	Pipe 2 SWB 2	SWB 1 Server 2	Unknown	SWB 1 Pipe 2 SWB 2
8	Pipe 2	1	CWP 2	SWB 1 Pipe 2 SWB 2	Unknown	CWP 2 SWB 1 SWB 2
9	CWP 2	1	SWB 2	CWP 2 SWB 1 SWB 2	Unknown	SWB 1 SWB 2

Network connectivity check:

- *Electric network*
  1. *Step 1 shows that server 1 is connected to SWB1*
  2. *Step 3 shows that CWP1 is connected to SWB1*
  3. *Step 4 shows that SWB1 is connected to SWB2*
  4. *Step 5 shows that SWB2 is connected to TF2*
  5. *Step 7 gives the connection between server 2 and SWB2*

     It can be seen that all of the users are connected to TF2 in this example
- *Chilled water network*
  1. *Step 1 shows that server 1 is connected to pipe 1*

2. *Step 2 shows that pipe 1 is connected to CWP1*
3. *Step 7 shows server 2 connected to pipe 2*
4. *Step 8 shows pipe 2 connected to CWP2*

   This network is also fully connected.

Example 4:
The final situation is a network in which the HUB-HUB connection has been hit. The damage to this network creates to parallel networks. Figure 4.13 shows the network, while table 4.7 shows the node investigation. The connectivity check is done in the end.

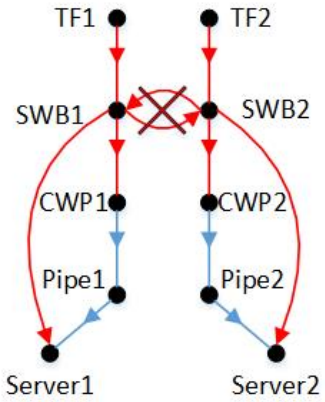


Figure 4.13: Example network of figure 4.10 with a HUB-HUB hit

Table 4.7: Node investigation, HUB-HUB hit example

Step	*Investigated node*	*Number of connected networks*	*Connected nodes*	*Old investigation list*	*Connected?*	*New investigation list*
1	*Server 1*	*2*	*Pipe 1* *SWB 1*	*Server 1* *Server 2*	*Unknown*	*Pipe 1* *SWB 1* *Server 2*
2	*Pipe 1*	*1*	*CWP 1*	*Pipe 1* *SWB 1* *Server 2*	*Unknown*	*CWP 1* *SWB 1* *Server 2*
3	*CWP 1*	*1*	*SWB 1*	*CWP 1* *SWB 1* *Server 2*	*Unknown*	*SWB 1* *Server 2*
4	*SWB 1*	*1*	*TF 1*	*SWB 1* *Server 2*	*Unknown*	*TF 1* *Server 2*
5	*TF 1*	*0*		*TF 1* *Server 2*	*Yes*	*Server 2*
6	*Server 2*	*2*	*Pipe 2* *SWB 2*	*Server 2*	*Unknown*	*Pipe 2* *SWB 2*
7	*Pipe 2*	*1*	*CWP 2*	*Pipe 2* *SWB 2*	*Unknown*	*CWP 2* *SWB 2*
8	*CWP 2*	*1*	*SWB 2*	*CWP 2* *SWB 2*	*Unknown*	*SWB 2*
9	*SWB 2*	*1*	*TF 2*	*SWB 2*	*Unknown*	*TF 2*
10	*TF 2*	*0*		*TF 2*	*Yes*	

Network connectivity check:

- *Electric network*

1. *Step 1 shows that server 1 is connected to SWB1*
2. *Step 3 shows that CWP1 is connected to SWB1*

3. *Step 4 shows that SWB1 is connected to TF1*
4. *Step 7 gives the connection between server 2 and SWB2*
5. *Step 9 shows a connection between CWP2 and SWB2*
6. *Step 5 shows that SWB2 is connected to TF2*

   The above connections show to parallel working networks (same as a fully connected network)

- *Chilled water network*

1. *Step 1 shows that server 1 is connected to pipe 1*
2. *Step 2 shows that pipe 1 is connected to CWP1*
3. *Step 7 shows server 2 connected to pipe 2*
4. *Step 8 shows pipe 2 connected to CWP2*

   This network is fully connected.

The four examples shown above, show the different scenarios that can exist during an investigation by this method. The above examples show the method for a single hit, however the exact same method is used when multiple hits are involved.

Vulnerability based on connectivity

Figure 4.9 shows that a vulnerability score is given after the analysis. This score is based on the input provided by the user, creating a wide variety of vulnerability scores for a single system topology based on the user input. This gives the user the freedom to not only determine which nodes to include in the investigation, but also to determine separate objectives to be included. To demonstrate the usefulness of extra objectives, a few examples of vulnerability scores are given.

- *Fully connected,*
  This concept is the simplest, the vulnerability score is only based on one criteria, namely; are all investigated nodes fully connected (e.g. they can receive all their required resource types)?

- *Fully connected with some redundancy,*
  A variation on the pure fully connected concept. In some cases, it is necessary to determine if redundancy is available. The redundancy can be found from the checklist as the number of suppliers connected is listed. For example, it is necessary to connect nodes to at least two suppliers, so that if one fails the node is still supplied.

- *Different states of the system,*
  Not every failure of an investigated node leads to failure of the entire system. This concept focusses on different states of the system based on which components are connected and which are not. For instance; a system topology of a naval vessel can include hotel functions and weapon systems. When the hotel functions are not connected (or they are off or in a low power state), but the weapon systems are; it could be concluded that the vessel could still fulfil its mission. However, by creating different states a single assessment it can be possible to determine the failure rates of both systems separately without doing two assessments.

4.4.4. Vulnerability function part II - Capacity

The second part of the tool uses the fully connected networks of the first part to create a better vulnerability assessment of the topology. The first analysis is focussed on the connections between the different components. The second assessment, discussed in this section determines whether components are still supplied as their resource requirements demand.

This section has the same build up as section 4.4.3 (part-I). First the input needed will be discussed followed by the method. With knowledge of the method examples will be given to illustrate the possible scenarios. Finally, the output of the function will be discussed.

Input

Most of the inputs needed by the second part of the tool are also needed by the first part. These inputs can be seen in section 4.4.3, but are also shown below (without elaboration) together with the new inputs with elaboration.

- *The distribution networks and the nodes that belong to them*
- *Possible connections*
- *The adjacency matrix*
- *Nodes for investigation*
- *Capacity of nodes,*
  The nodes need a certain amount of input to generate a certain amount of output. This should be given as an input to the tool. The capacity of the nodes will be given in this research in power [W]. The capacity could be given in the more general effort and flow variables (effort × flow = power), but this would introduce time dependant effects into the method which are not taken into account. So, for each node the following information should be provided.
  1. *Amount needed of a certain resource (input)*
  2. *Amount generated of a certain resource (output)*
  3. *Factor linking input to output, this factor will state how the input and output of a component influence each other.*
- *Capacity of the edges,*
  The edges within the network are in most cases not able to transport an unlimited amount of a certain resource. To consider this the upper and lower bounds of every edge should be given with respect to the amount of resources it can transport at any given moment.
- *Objective function*
  During the method, the objective function will be elaborated. However, it can be stated that the objective function is needed to solve the problem. It is this function that determines how the problem should be solved. Is it necessary to maximize the output of a distribution system, or should the flow through some edges be minimized?

With these new inputs the method can be described.

The method

To solve the capacity problem of a given network a linear programming or linear optimization will be used [Hillier and Lieberman, 2010]. The linear optimization method has been chosen, as it can be used to determine the optimal flow through a network. Using this optimal flow, it can be concluded whether the prime users are supplied with enough resources to function. A list can be formed containing all the prime users together with their respective supplied resource level x with $0\% \leq x \leq 100\%$. This resource level can then be used to define if the network is still working. The vulnerability of the topology is than taken as the chance of the prime users in a topology not being supplied.

How does it work?

The starting point is the determination of the prime users. These are the nodes that will be examined during this method to determine the vulnerability based on the networks capacity. To determine the amount of resources supplied to the prime users, the entire network is split into several smaller networks, the distribution networks. The flow within these distribution networks is determined using a linear optimization algorithm. From the flow, it can be found which users, in each network, are supplied, and by how much of a certain resource they are supplied. The supply to these users can then be used to determine the output of the users to the distribution networks that they supply. Following this trend from the bottom of the network (Main Suppliers) to the top (prime users) gives an insight into the state of the network. The state of the network could be a whether a certain component is supplied correctly, in this case the state is either zero or one depending on the supply to the component.

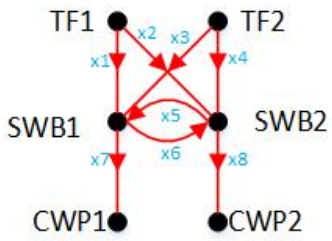
The linear optimization algorithm
Determining the flow through a distribution network is accomplished using a linear optimization algorithm. The algorithm used is based on maximizing the flow toward the users of a distribution network. This maximization ensures that the maximum amount of resources is sent through the network as a flow. To describe the algorithm used, the first step is to explain the basics of a linear optimization algorithm. Secondly the basics will be used to determine the form of the linear optimization as used during this research.

A linear optimization uses an objective function and one or more constraints to describe the optimization problem. The objective function is used to determine what to maximize or minimize. The constraints determine the range in which the objective function has to be found. Within the constraints, two groups can be separated:

1. *Problem constraints*
2. *Non-negativity constraints*

Problem constraints determine specific constraints with respect to the problem these constraints could be: equilibrium of in- and outputs of nodes, maximum amount of output of a node, or maximum amount of flow through an edge. The non-negativity constraints determine that no variable used can be smaller than 0. The structure of a linear optimization problem is shown in example 5 besides figure 4.14.

Example 5:
A simple $440V$ *distribution network is supplied by two transformers (*$TF1$ *and* $TF2$*) with a total of* $2kW$ *per transformer. The electricity generated by the transformers is led through two switchboards (*$SWB1$ *and* $SWB2$*) towards two chilled water plants (*$CWP1$ *and* $CWP2$*). The chilled water plants have a capacity of* $1.5kW$*. The optimization of this system will be done by maximizing the flow through the two edges leading to the chilled water plants.*



Objective function:	
$max:$	$Z = x_7 + x_8$
Problem constraints:	
$s.t.$	$x_1 + x_2 \leq 2$ $x_3 + x_4 \leq 2$ $-x_1 - x_3 - x_5 + x_6 + x_7 = 0$ $-x_2 - x_4 + x_5 - x_6 + x_8 = 0$ $-x_7 \geq -1.5$ $-x_8 \geq -1.5$
Non-negativity constraints:	
$with,$	$\boldsymbol{x} \geq 0$

Figure 4.14: A $440V$ distribution network with flow parameters x

The above example is already written in the form needed to give a clear description of the distribution network shown in figure 4.14 How this form originates from this distribution network will be described for every of the three parts of the problem: the objective function, the problem constraints, and the non-negativity constraints.

1. *Objective function,*
   The objective function is used to determine the maximum flow through the network shown in figure 4.14. There are three methods to maximize this function, these methods are:
   (a) Taking the sum of every variable x

$$Z = \sum_{i=1}^{n} x_i, \qquad 1 \leq i \leq n \tag{4.15}$$

   This form has a single disadvantage, the maximum value is only gotten if the edges between HUBs are fully utilized. For example 5, this would mean that even though TF1 can supply CWP1 and TF2 can supply CWP2 the maximum value would only be reached if x_5 and x_6 are maximized.

(b) Taking the sum of all edges outgoing from the suppliers

$$Z = \sum_{i=1}^{n} x_i, \qquad 1 \leq i \leq n_{supplier\ edges} \tag{4.16}$$

(c) Taking the sum of all edges going to the users

$$Z = \sum_{i=1}^{n} x_i, \qquad 1 \leq i \leq n_{user\ edges} \tag{4.17}$$

Both the second and the third method are good choices since they do not have any downsides. In the end, it was decided to use method 3. This method was chosen as it fits into the style used by the MATLAB linprog-function. The linprog-function is made to only handle minimization objectives. Method 2 could have been made into a minimization method, however the values for method 3 are already available.

Using the objective function for the linear programming problem, there is another part that should be looked into. Some users are more favourable to be supplied with energy due to connection to the prime users in higher distribution layers. To show this concept, the importance factor (IF) is introduced. This factor is given to a prime user, showing the importance of that user. During the connectivity check, the importance factor of a prime user is given to each component in its connection tree when no higher importance factor is present. Using this build up, the importance of components can be shown in their respective linear programming problem by multiplying their variables x with the IF. This gives higher priority to the users connected to the prime users assuring that the resources reach the prime users.

2. *Problem constraints,*
To determine the flow through a distribution network it is necessary to create a framework of rules that govern the problem. This framework is shown in the problem constraints.

The first step is to show the basics of going to a linear optimization form from the network form. Example 5 uses different variables x to depict which of the edges is meant. To create a standard approach to convert the network to a linear optimization, the generation of the x variables should be constant for every conversion. Figure 4.15 depicts a simple four nodes directed network. This figure also shows the distribution of the x variables within this network. As it can be seen in the figure, the first aspect to determine the x variable is the starting node of the edge. Thus, node 1 will be the first node under investigation as starting point followed by node two etc. For this node, all other nodes that it has a connection with are considered, ranking these nodes from low to high gives the ranking of the x variables. This can be seen in figure 4.15 as node 1 has an outgoing connection to both node 2 and node 4. The edge towards node 2 is the first edge to be considered ($1 \rightarrow 2$) followed by node 1 to node 4 ($1 \rightarrow 4$) thus $x_{1\rightarrow 2} = x_1$ and $x_{1\rightarrow 4} = x_2$.

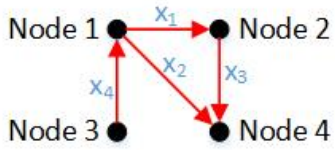


Figure 4.15: example of parameter x on a 4 nodes directed network

When the x variables are distributed over all of the directed edges, the next step is to determine when a variable is either negative or positive when written within the problem constraints. The difference between positive and negative is based on the direction of the edge and the capacity of the components. A supplier will supply energy to a system, thus the supplier has a positive amount of energy. A user is asking energy from a system and thus has a negative amount of energy. To keep the signs within the problem constraints homogeneous, outgoing edges will be taken positive while incoming edges are taken negative. This can also be seen in figure 4.16.

With the sign convention and the assignment of x variables to the edges, the last step is to determine how each node should be expressed. Since there are three different kinds of nodes, there are also three different ways to describe them.

Figure 4.16: Positive and negative flow of a linear program shown in graph form

(a) *Supplier,*
Since the supplier supplies energy, the amount of energy E will be taken positive. As by the sign convention an outgoing edge is also positive. Thus, a constraint based on a supply node will only have positive x variables and a positive amount of energy needed. Finally, a supplier can never supply more energy than the maximum amount given. With these the following equation can be made describing a supply node constraint.

$$\sum x_{out} \leq E \tag{4.18}$$

(b) *HUB,* A HUB is needed to distribute energy between different components. A HUB will by itself not take in any energy. Thus, the amount of energy coming in and going out of a HUB should be in equilibrium.

$$\begin{aligned} \sum x_{out} &= \sum x_{in} \\ \sum x_{out} - \sum x_{in} &= 0 \end{aligned} \tag{4.19}$$

(c) *User,*
The users are the opposite of the suppliers, they take energy out of the system. The edges connected to a user will always supply the user and should be taken negative.

$$\begin{aligned} -\sum x_{in} &\geq -E \\ \sum x_{in} &\leq E \end{aligned} \tag{4.20}$$

The above three equations can also be seen in example 5 where they are written in their entire length. The choice to keep the equation for the user constraint as shown in the example, was made to clearly reflect whether a constraint was based on a supplier or a user. Describing these two constraints (supplier and user) in the same manner could result in unclear problem constraints for larger networks.

Some components can only function if they are supplied with atleast a minimum amount of energy, this can be shown using the constraints as shown above by adding a second constraint for the users/suppliers with this minimum amount constraint. This is done by reversing the greater or equal sign and adding a percentage to the number of resources. This is shown below for a user.

$$\begin{aligned} \sum x_{in} &\leq E \\ \sum x_{in} &\geq \%E \end{aligned} \tag{4.21}$$

3. *Non-negativity constraints*
The last constraints to identify before being able to completely write down the linear optimization, are the non-negativity constraints. These constraints give the boundaries for the x variables or edges. The non-negativity constraint is used to keep a directed edge directed. For instance, when a non-negative constraint is missing it could be possible that an edge would become negative. A negative edge shows a reverse flow and should be impossible.

   Since the non-negativity constraint is the only constraint that can freely dictate the state of the edges, these constraints will also be used to indicate these states. The easiest way to accomplish this is to provide the problem with a maximum flow through an edge. This example can easily be brought into practice, as a pipe will not be able to generate an unlimited flow. At a certain point the flow is too fast to accelerate even more and will thus stay at this maximum point. Another edge state that could be described is a minimum amount of energy needed to create a flow.

Using the above information, it can now be seen why the linear optimization problem of example 5 has its current form. To elaborate further on this method, the following section will show different scenarios on which the method can be used. But first, the final part of the method will be explained.

How to determine if a user is supplied correctly?
This final piece of the method will elaborate on the evaluation of users to determine whether they work or not. As stated before the prime users are the nodes that should be examined. A prime user can be any node within the network but will most likely be a user in one of the highest distribution networks. To evaluate the availability of a prime user it is necessary to determine whether it is supplied with enough resources. To accomplish this, a loop is generated that determines which distribution networks are ready for examination.

The first step of the loop is to determine which of the distribution networks has main suppliers MS, these suppliers don't need any resources themselves and can thus be used as the starting point of the loop. After finding the flow through the initial distribution network, the next distribution network that is supplied will be sought. This process continues until all distribution networks have been examined or until all the prime users have been examined, whichever comes first.

The loop is simple, but will give some restrictions with respect to the sort of system that can be explored. The following networks cannot be examined because of the current restrictions to the availability within a network.

- *Networks with dependencies,*
  Dependencies here means components that need to be supplied by components that they supply. For instance, a diesel generator needs cooling water, air and fuel to work. The diesel generator produces electricity. The electricity is used to propel some components of which three are pumps. These pumps are used to pump cooling water, air and fuel to the diesel generator. This is called a dependency. The pumps depend on the generator for electricity and the generator depends on the pumps for his supply.
- *Networks that have two types of suppliers (main and normal),*
  The problems occurring with these networks is that they could be examined before all suppliers are given resources, thus reducing the amount of available flow in the network.

Scenarios

This section will be dedicated to different scenarios that can occur during the use of this method. These scenarios will all be explained and it will be shown how an answer is found in all the given scenarios. The network used is the same two-layer network as shown in section 4.4.3 in the examples, but the two pipes are connected. The new two-layer network can be seen in figure 4.17. The information of the nodes within this network is shown in table 4.8. Besides the shown network and node information the following information is

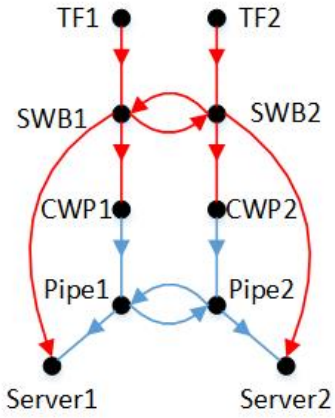


Figure 4.17: Two layer network to be used in capacity examples

needed for the scenarios:

- The system has two servers: Primary server (server 1) and a secondary server (server 2). The primary server should always be provided with power, while the secondary server can be shut down. The primary server has thus an importance factor (IF) of 2 while the secondary server has an IF of 1. Both servers can fully operate at 60% of their installed capacity, however when this limit is not reached for the secondary server it will be shut down. The chilled water plants can always operate, with an output equal to their input.

- The solution to the state of the servers will be given as a 1 when the server is working and a 0 when the server is not working, thus $state_{server(x)} = [0\ 1]$.

Table 4.8: Node information of the network shown in figure 4.17

Name	Node Nr.	Network	Job	Capacity
TF1	1	440V	Supplier	30 kW
TF2	2	440V	Supplier	30 kW
SWB1	3	440V	HUB	-
SWB2	4	440V	HUB	-
CWP1	5	440V	User	-15 kW
		CW	Supplier	20 kW
CWP2	6	440V	User	-15 kW
		CW	Supplier	20 kW
Pipe1	7	CW	HUB	-
Pipe2	8	CW	HUB	-
Server1	9	440V	User	-15 kW
		CW	User	-15 kW
Server2	10	440V	User	-15 kW
		CW	User	-15 kW

Scenario 1: A fully connected network
The first scenario is the fully connected network as seen in figure 4.17. The problem can be reduced to the objective function and constraints as shown below. First the adjacency matrix, see figure 4.18a, is used to determine the variables, see figure 4.18b. From the variables and the rest of the known information, the linear problem can be created, see table 4.10.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) The adjacency matrix of figure 4.17

$$A = \begin{bmatrix} 0 & 0 & x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & x_4 & 0 & 0 & 0 & x_5 & 0 \\ 0 & 0 & x_6 & 0 & 0 & x_7 & 0 & 0 & 0 & x_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & x_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{13} & 0 & 0 & x_{14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Variables from the adjacency matrix of figure 4.18a

Figure 4.18: The adjacency matrix and variables of the fully connected network

After creating the linear problem, the next step is to solve it. Using the values for the constraints from table 4.8, the linear problem from table 4.9 becomes as shown in table 4.10. Solving the above problem gives the solution as seen in 4.11 and illustrated in figure 4.19. This solution is just a single solution of a set of solutions. This set occurs since the program is free to use the HUBs as it pleases. Thus, there will be many solution in which the HUBs transport some amount of flow between them.

The solution of this scenario shows that both servers are working, since both states of the servers are one. This state can easily be checked, $server1$ is fed by x_5 $(440V)$ and x_{12} (CW), and $server2$ is fed by x_8 $(440V)$ and x_{14} (CW). All of these parameters have a value of 15 which is the maximum amount of energy the servers can take as an input.

Table 4.9: Linear programming problem of network in figure 4.17

	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$x_1 \leq E_{TF1}$	$x_9 \leq E_{CWP1}$
	$x_2 \leq E_{TF2}$	$x_{10} \leq E_{CWP2}$
	$-x_1 + x_3 + x_4 + x_5 - x_6 = 0$	$-x_9 + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-x_4 \geq -E_{CWP1}$	$-x_{12} \geq -E_{server1}$
	$-x_7 \geq -E_{CWP2}$	$-x_{12} \leq -E_{server2} * 60\%$
	$-x_5 \geq -E_{server1}$	$-x_{14} \geq -E_{server2}$
	$-x_5 \leq -E_{server1} * 60\%$	
	$-x_8 \geq -E_{server2}$	
n.n.c.	$\boldsymbol{x \geq 0}$	$\boldsymbol{x \geq 0}$

Table 4.10: Complete linear programming problem scenario 1

	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$x_1 \leq 30$	$x_9 \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-x_1 + x_3 + x_4 + x_5 - x_6 = 0$	$-x_9 + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-x_4 \geq -15$	$-x_{12} \geq -15$
	$-x_7 \geq -15$	$-x_{12} \leq -9$
	$-x_5 \geq -15$	$-x_{14} \geq -15$
	$-x_5 \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$\boldsymbol{x \geq 0}$	$\boldsymbol{x \geq 0}$

Table 4.11: Solution to the problem of table 4.10

440V Network	***Cooling Water Network***
$\boldsymbol{Z_{440V} = 105}$	$Z_{CW} = 45$
$\boldsymbol{x_1 = 30}$	$x_9 = 15$
$\boldsymbol{x_2 = 30}$	$x_{10} = 15$
$\boldsymbol{x_3 = 0}$	$x_{11} = 0$
$\boldsymbol{x_4 = 15}$	$x_{12} = 15$
$\boldsymbol{x_5 = 15}$	$x_{13} = 0$
$\boldsymbol{x_6 = 0}$	$x_{14} = 15$
$\boldsymbol{x_7 = 15}$	
$\boldsymbol{x_8 = 15}$	

$$\boldsymbol{state_{server1} = 1}$$
$$\boldsymbol{state_{server2} = 1}$$

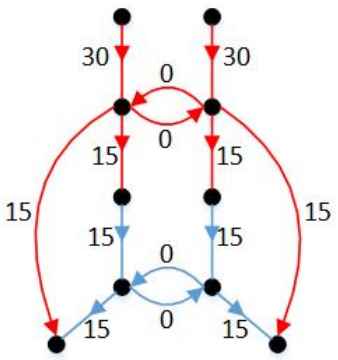
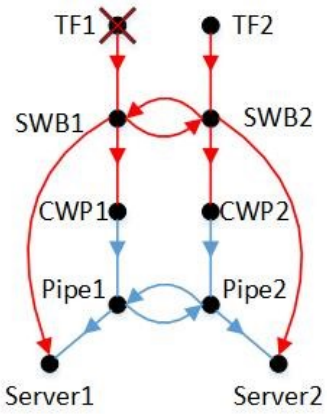


Figure 4.19: Flow solution of scenario 1

Scenario 2: A main supplier is hit

The second scenario shows how the function reacts when a main supplier is hit. The damaged graphical network and its matrices can be seen in figure 4.20. Table 4.12 shows the linear programming problem for this scenario which follows from figure 4.20 and table 4.9. The solutions for this scenario can be seen in table 4.13 and the graphical representation in figure 4.21.

In this situation, only the primary server is working since it is supplied with at least 60%. It can also be seen that it is impossible to get the secondary server running as $4 \times 9 = 36kW$ $(440V)$ is needed, while only $30kW$ $(440V)$ is provided. The given solution is one of the possible solutions that lead to a working primary server, while the secondary server is offline. Although half of the total capability of the network is lost, it still satisfies the constraints



(a) Network graph

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(b) Adjacency matrix

$$A = \begin{bmatrix}
0 & x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & x_3 & x_4 & 0 & 0 & 0 & x_5 & 0 \\
0 & x_6 & 0 & 0 & x_7 & 0 & 0 & 0 & x_8 \\
0 & 0 & 0 & 0 & 0 & x_9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x_{10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x_{11} & x_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & x_{13} & 0 & 0 & x_{14} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(c) Variables

Figure 4.20: Initial information of scenario 2

Table 4.12: Complete linear programming problem scenario 1

	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$0 \leq 30$	$x_9 \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-x_1 + x_3 + x_4 + x_5 - x_6 = 0$	$-x_9 + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-x_4 \geq -15$	$-x_{12} \geq -15$
	$-x_7 \geq -15$	$-x_{12} \leq -9$
	$-x_5 \geq -15$	$-x_{14} \geq -15$
	$-x_5 \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$\boldsymbol{x} \geq 0$	$\boldsymbol{x} \geq 0$

Table 4.13: Solution to the problem of table 4.12

440V Network	Cooling Water Network
$Z_{440V} = 60$	$Z_{CW} = 45$
$x_1 = 0$	$x_9 = 10$
$x_2 = 30$	$x_{10} = 10$
$x_3 = 0$	$x_{11} = 0$
$x_4 = 10$	$x_{12} = 15$
$x_5 = 10$	$x_{13} = 0$
$x_6 = 0$	$x_{14} = 5$
$x_7 = 10$	
$x_8 = 0$	

$$state_{server1} = 1$$
$$state_{server2} = 0$$

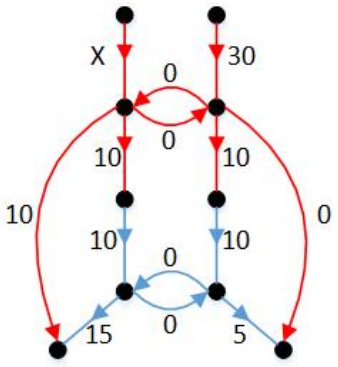
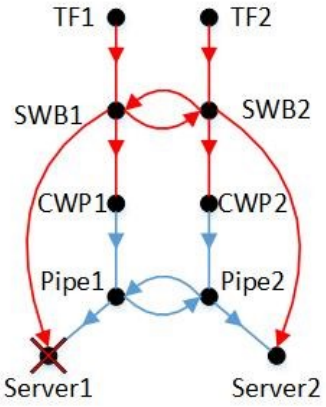


Figure 4.21: Flow solution of scenario 2

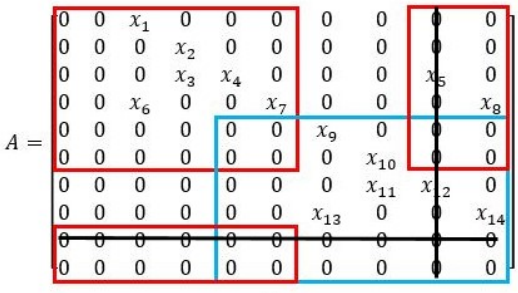
Scenario 3: A prime user is hit

The third scenario is used to show how the program would react to the failure of a prime user. In this scenario, server1 is hit as shown in the graphical form in figure 4.22a. Figure 4.22b and 4.22c show the matrices corresponding to the network of figure 4.22a. The linear programming problem of this scenario is given in table 4.14.



(a) Network graph

(b) Adjacency matrix



(c) Variables

Figure 4.22: Initial information of scenario 3

For the problem in table 4.14, no solution is shown as no solution exists. In both networks one of the constraints is not met, since the hit in server1 makes the resulting constraint impossible. The impossible constraints are:

- $$440V: \ -(0) \leq -9, \quad untrue$$
- $$CW: \ -(0) \leq -9, \quad untrue$$

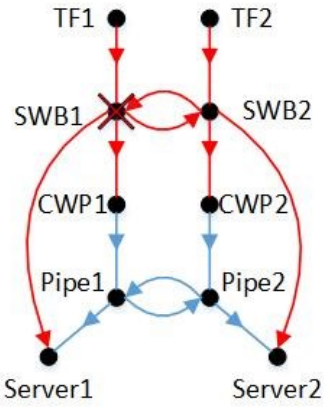
As no solution is possible, this scenario will lead to a system capability of 0%. If for instance both servers would have had the same importance, the capability of the system would have been at 50%.

Table 4.14: Complete linear programming problem scenario 3

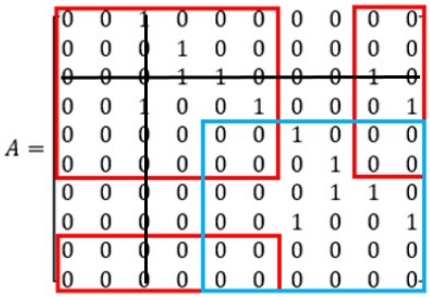
	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2(0) + 2x_7 + x_8$	$Z_{CW} = 2(0) + x_{14}$
s.t.	$x_1 \leq 30$	$x_9 \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-x_1 + x_3 + x_4 + x_5 - x_6 = 0$	$-x_9 + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-x_4 \geq -15$	$-(0) \geq -15$
	$-x_7 \geq -15$	$-(0) \leq -9$
	$-(0) \geq -15$	$-x_{14} \geq -15$
	$-(0) \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$\boldsymbol{x} \geq 0$	$\boldsymbol{x} \geq 0$

Scenario 4: A HUB is hit

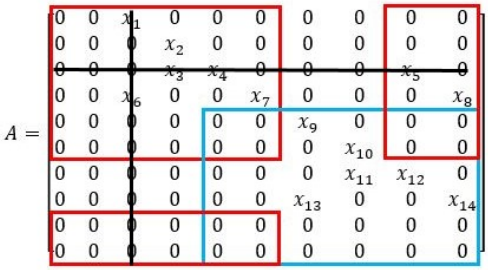
The fourth scenario shows the failure of a HUB node, as shown in the graphical network of figure 4.23a. The matrices depicting this failure are shown in figure 4.23b and figure 4.23c. The linear programming problem is shown in table 4.15.



(a) Network graph



(b) Adjacency matrix



(c) Variables

Figure 4.23: Initial information of scenario 4

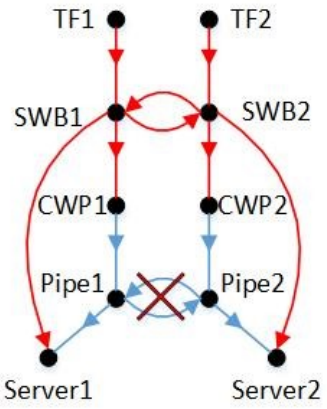
The programming problem shown in table 4.15 has no solution. As was the case in scenario 3, this scenario also has an impossible constraint. This constraint is again based in the $440V$ distribution network, and shows that the primary server ($server1$) is separated from its electrical power supply due to the failure of the HUB. The constraint that is impossible to meet is, $-(0) \leq -9$. This scenario leaves a capability of 0% for the system as no solution can be found.

Table 4.15: Complete linear programming problem scenario 4

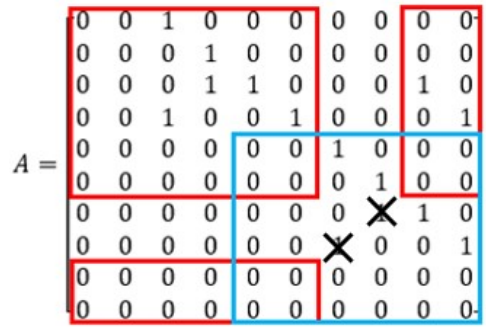
	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$(0) \leq 30$	$x_9 \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-(0) + (0) + (0) + (0) - (0) = 0$	$-x_9 + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - (0) + (0) + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-(0) \geq -15$	$-x_{12} \geq -15$
	$-x_7 \geq -15$	$-x_{12} \leq -9$
	$-(0) \geq -15$	$-x_{14} \geq -15$
	$-(0) \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$\boldsymbol{x} \geq 0$	$\boldsymbol{x} \geq 0$

Scenario 5: A HUB-HUB connection is hit

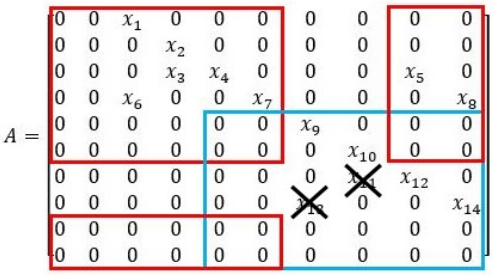
In the fifth scenario a HUB-HUB connection is hit. This is shown in figure 4.24a and the corresponding matrices in figure 4.24b and figure 4.24c. Table 4.16 shows the linear programming problem and the solution is shown in table 4.17. The flow network is shown in figure 4.25.



(a) Network graph



(b) Adjacency matrix



(c) Variables

Figure 4.24: Initial information of scenario 5

Table 4.16: Complete linear programming problem scenario 5

	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$x_1 \leq 30$	$x_9 \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-x_1 + x_3 + x_4 + x_5 - x_6 = 0$	$-x_9 + (0) + x_{12} - (0) = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - (0) + (0) + x_{14} = 0$
	$-x_4 \geq -15$	$-x_{12} \geq -15$
	$-x_7 \geq -15$	$-x_{12} \leq -9$
	$-x_5 \geq -15$	$-x_{14} \geq -15$
	$-x_5 \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$\boldsymbol{x \geq 0}$	$\boldsymbol{x \geq 0}$

Table 4.17: Solution to the problem of table 4.16

440V Network	***Cooling Water Network***
$\boldsymbol{Z_{440V} = 105}$	$Z_{CW} = 45$
$\boldsymbol{x_1 = 30}$	$x_9 = 15$
$\boldsymbol{x_2 = 30}$	$x_{10} = 15$
$\boldsymbol{x_3 = 0}$	$x_{11} = 0$
$\boldsymbol{x_4 = 15}$	$x_{12} = 15$
$\boldsymbol{x_5 = 15}$	$x_{13} = 0$
$\boldsymbol{x_6 = 0}$	$x_{14} = 15$
$\boldsymbol{x_7 = 15}$	
$\boldsymbol{x_8 = 15}$	

$$\boldsymbol{state_{server1} = 1}$$
$$\boldsymbol{state_{server2} = 1}$$

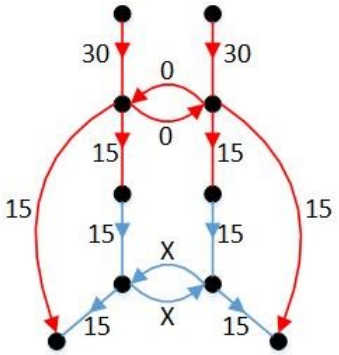


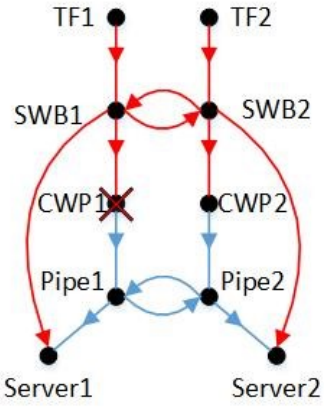
Figure 4.25: Flow solution of scenario 5

The solution shown in table 4.17 determines that no capability is lost due to a hit to the HUB-HUB connection. In this case, the only thing that has happened is that the chilled water network has changed to two disconnected distribution lines.

Scenario 6: A converter is hit

The final scenario shows how the system would react when a converter is hit. This scenario also shows the importance of the HUBs and their task to reconfigure a system once a supplier or user is hit. The sixth scenario can be seen in figure 4.26a with its corresponding matrices in figure 4.26b and 4.26c. The entire linear programming problem is seen in table 4.18 and the solution is shown in table 4.19 together with the flow network in figure 4.27.

For the converter hit, the solution of table 4.19 shows a 50% capable system. This may come as a surprise, as there is enough power available to keep both server units running. The decision to power off the secondary server was made due to the constraints given, which gave the primary server a higher importance factor.



(a) Network graph

$$A = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(b) Adjacency matrix

$$A = \begin{bmatrix}
0 & 0 & x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & x_3 & x_4 & 0 & 0 & 0 & x_5 & 0 \\
0 & 0 & x_6 & 0 & 0 & x_7 & 0 & 0 & 0 & x_8 \\
0 & 0 & 0 & 0 & 0 & 0 & x_9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & x_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x_{13} & 0 & 0 & x_{14} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(c) Variables

Figure 4.26: Initial information of scenario 6

Table 4.18: Complete linear programming problem scenario 6

	440V Network	***Cooling Water Network***
Max:	$Z_{440V} = 2x_4 + 2x_5 + 2x_7 + x_8$	$Z_{CW} = 2x_{12} + x_{14}$
s.t.	$x_1 \leq 30$	$(0) \leq 20$
	$x_2 \leq 30$	$x_{10} \leq 20$
	$-x_1 + x_3 + (0) + x_5 - x_6 = 0$	$-(0) + x_{11} + x_{12} - x_{13} = 0$
	$-x_2 - x_3 + x_6 + x_7 + x_8 = 0$	$-x_{10} - x_{11} + x_{13} + x_{14} = 0$
	$-(0) \geq -15$	$-x_{12} \geq -15$
	$-x_7 \geq -15$	$-x_{12} \leq -9$
	$-x_5 \geq -15$	$-x_{14} \geq -15$
	$-x_5 \leq -9$	
	$-x_8 \geq -15$	
n.n.c.	$x \geq 0$	$x \geq 0$

Table 4.19: Solution to the problem of table 4.18

440V Network	***Cooling Water Network***
$Z_{440V} = 75$	$Z_{CW} = 35$
$x_1 = 15$	$x_9 = 0$
$x_2 = 30$	$x_{10} = 20$
$x_3 = 0$	$x_{11} = 0$
$x_4 = 0$	$x_{12} = 15$
$x_5 = 15$	$x_{13} = 0$
$x_6 = 0$	$x_{14} = 5$
$x_7 = 15$	
$x_8 = 15$	

$$state_{server1} = 1$$
$$state_{server2} = 0$$

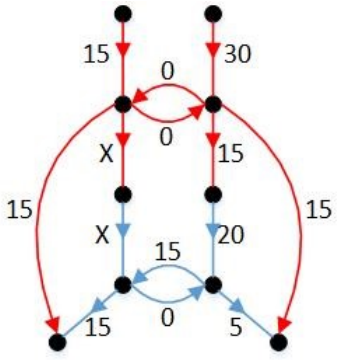


Figure 4.27: Flow solution of scenario 6

Vulnerability based on capacity

The output of part-II of the vulnerability function determines the objective function: vulnerability, based on the capacity of the system. As shown in the scenarios of the previous section, the outcome of the vulnerability can be steered by the constraints that are given to the function and the degree of importance of certain nodes. Using these steering mechanisms together with the identification of different states of certain nodes allows to look at different situations at the same moment ,or to look at the capability based on the prime users and the availability of the less important users. This differentiation for a frigate that should be capable to run an air defence mission, could be:

- *Prime users*
  - Sensor systems
  - Data analysation systems
  - Air defence systems
- *Other users*
  - Hotel systems
  - Other weapon systems

So, for every damaged network it would be possible to determine whether the vessel is capable to complete its primary mission or not. But, it can also be seen to what extent the vessel is capable to fulfil different functions.

4.4.5. Assessment of the method

The vulnerability assessment method described above has its advantages and disadvantages. This section will assess the method based on the advantages and disadvantages shown below. The assessment will give a short summary on the developed method as well as describing the problems that still exist.

Advantages:

- Two levels of detail can give either an indication of the vulnerability (part-I) or a complete estimation (part-II).
- Part-I is fast in respect to part-II, thus the answer of part-I is used to filter the networks that are not connected.
- The method uses modelled damage to the networks to assess the reaction of the network to said damage. Due to this behaviour, the vulnerability of the networks will be estimated instead of predicted or weighed against each other with use of their characteristics, as was the case with the metrics of section 4.2 and 4.3.
- The sample size determination using the normal distribution gives a maximum number of needed samples that is far smaller than the actual population (for larger networks or more hits) reducing the time needed to calculate the vulnerability.
- By defining different prime users, different states of the system can be evaluated at the same time. However, evaluating different states may lead to a generation of a non binary problem which leads to an increased calculation time.

Disadvantages:

- When the vulnerability based on the capacity is added to the method, the amount of time needed to generate results increasing drastically.
- The method cannot handle two nodes that are dependent on each other (e.g. a diesel generator asking chilled water generated by a chilled water plant needing the electricity generated by the generator). Such a dependency leads to a infinite loop.
- The chance that failure occurs is uniform for every node and edge.

- For more complex networks, a problem may occur with the determination of the flows of energy. Precautions have been implemented, but for the more complex networks a misdirected flow of energy can be possible.

The strength of this method lies in the separation of an less accurate indication of the vulnerability and an accurate calculation based on the availability of resources to selected components. The first part will filter the disconnected topologies from the group indicating if networks will have a chance to work after failure. The second part than evaluates the still connected topologies to find the rest of the failing systems. Since the method uses modelled damage, the calculations can be done and tell something about the behaviour of the system after failures have occurred. However, the modelling of the failures is still not close to reality as the chance of failure for every given component/connection in the system is uniform. This uniformity describes the randomness of the hits, but some parts of the vessel or some components will be more inclined to failure than others.

The downside to the method is that it takes time to find the correct answers and it would drastically increase when higher accuracy is demanded. Also the fact that the method cannot handle dependant nodes creates a problem when looking into networks with these connections.

4.5. Conclusion

This chapter showed the different forms of vulnerability prediction of network topologies. The metrics found in literature are not capable to deal with the system logic introduced by the topology generator as discussed in chapter 3. The problems found within these metrics helped to create the vulnerability function as shown in section 4.4. The maxflow method described in section 4.3 uses the basic system logic to assess the system topologies. Although the method can tell a lot about a system topology, it has some undesired outcomes. For instance, the fully connected HUBs while the users and suppliers have only one connection per input and output. But the maxflow method showed how the HUBs are an important part of the entire network with respect to the configurative ability of these HUBs.

Section 4.4 shows the function created during this research. This function uses damaged system topologies to assess the vulnerability of a topology. Using the actual damage of a system it is assumed that a higher level of accuracy of the vulnerability can be gotten.

5

Verification of the tool and results

This chapter will describe the application and results of the vulnerability assessment method. The chapter is divided into three sections. First, the results generated by the tool are verified. This is done by comparing the results of the tool with hand calculated results for some simple multi-layered networks. The second part evaluates the changes in outcome off the tool for different forms of a single distribution layer, to identify if the behaviour with respect to the vulnerability of the layer is as expected. The last part of this chapter shows the results of the method for a test case. This case is based on the network for a notional frigate. The answers found by the tool for this network is then used to determine how a concept design of this network performs. Does the current way of design give a good solution?

5.1. Tool verification

The first step into the application and results of the tool is to verify the results generated by the tool. To make a good verification, it is necessary to use a multi-layered network. A multi-layered network can be generated in such a manner, that it consists of all possible connection options for a network generated by SDS-ATG. The network used for the verification is described in section 5.1.1 Using the described network, four random networks are generated using SDS-ATG. These four networks are then submitted to a vulnerability assessment by the tool as well as an hand calculation to verify the tools answer. The results of these calculations and exact answers are shown in section 5.1.2.

5.1.1. Description of the network

The network used to verify the method is a simple two-layer network. The two distribution networks used are the $440V$ network and the chilled water network, these two layers can be seen fully connected in figure 5.1. The names and abbreviations of the nodes together with their network type, capacity and minimum required amount of input can be seen in table 5.1. The capacities and minimum required inputs are based on the values of a notional frigate.

Figure 5.1: Distribution layers of a two-layer network (fully connected) with $440V$ (left) and chilled water (rigth)

The network described in table 5.1 has been separated into two zones, all the components with a 1 in their name are within the first zone, the other components are located in the second zone. Each zone has its own server, these are the prime users of the network and are both needed with at least their minimum input to

Table 5.1: Node data for the network layers shown in figure 5.1

Node	**Abbreviation**	**Network type**	**Capacity** $[kW]$	**Min. capacity req.**
Transformer 1	TF1	$440V$	200	-
Transformer 2	TF2	$440V$	200	-
Switchboard 1	SWB1	$440V$	0	-
Switchboard 2	SWB2	$440V$	0	-
Chilled water plant 1	CWP1	$440V$	-75	-
		CW	200	-
Chilled water plant 2	CWP2	$440V$	-75	-
		CW	200	-
Pipe system 1	Pipe1	CW	0	-
Pipe system 2	Pipe2	CW	0	-
Server 1 1	Ser1	$440V$	-125	60%
		CW	-200	60%
Server 2 2	Ser2	$440V$	-125	60%
		CW	-200	60%

have a system that works. In other words, the prime users have the same importance in the network, so no importance factor is used. In this case, only two states exist for the vulnerability assessment, as seen in equation 5.1.

$$state_{network} = \begin{cases} 0, & if\ both\ servers\ function \\ 1, & if\ one\ OR\ both\ server(s)\ fail(s) \end{cases} \tag{5.1}$$

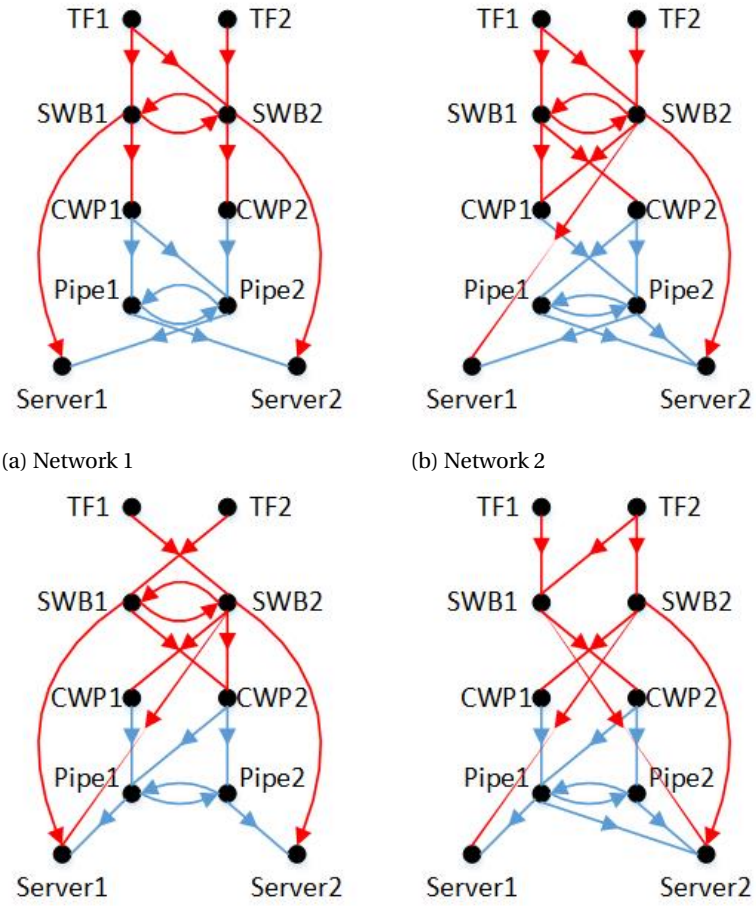
5.1.2. Results

The chosen network contains all the elements that could occur within a topology. Besides having all these elements, the chosen network is also small enough (10 nodes and 24 edges) to allow an exact approach of the used methods. The small scale of the network gives the option to calculate every possibility of hits for the given network when no more than two hits are considered. Using every possibility, the tool calculations ensures that the tool makes the right decision for every possibility. Table 5.2 shows the vulnerabilities found for four different networks shown in figure 5.2.

Vulnerability of four random two-layer networks								
	Part I - Connectivity				Part II - Capacity			
	1 Hit		2 Hits		1 Hit		2 Hits	
Network	Tool [%]	Exact [%]	Tool [%]	Exact [%]	Tool [%]	Exact [%]	Tool [%]	Exact [%]
1	38,4	38,4	68,6	68,6	69,2	69,2	93,2	93,2
2	25	25	47,4	47,4	53,6	53,6	81,5	81,5
3	29,6	29,6	56,7	56,7	63,0	63,0	88,6	88,6
4	30,8	30,8	56,9	56,9	61,5	61,5	87,7	87,7

Table 5.2: Vulnerability results of a simple two-layer network

The results shown in table 5.2 verify the solutions found by the tool. In all cases, the vulnerability calculated by the tool is equal to the exact solution. The exact solutions can also be found in appendix B. With the results, it is shown that the vulnerability tools give the correct answer with respect to both parts of the vulnerability assessment for multi-layered network topologies for different hit cases.



(a) Network 1

(b) Network 2

(c) Network 3

(d) Network 4

Figure 5.2: Random two-layer networks used for the verification of the tool, see table 5.2

5.2. Behaviour of a single-layer distribution network

With the verification of the correctness of the solutions found by the tool, the next step is to look at the behaviour of the tool for different networks. This section considers single-layer distribution networks. These networks can be used to describe the individual behaviour of distribution layer with respect to the flow through a layer and the total amount of connections within the layer. Using the behaviour of the single-layer distribution networks, the influence of these networks on an entire network topology can be examined. The first part of this section describes the single-layer networks used to identify the behaviour of the distribution layers. The second part shows the results of these networks as the amount of flow that they can transport from the users to the suppliers. The flow will be used to determine the effect of amount of users/suppliers/HUBs/connections on the overall robustness of a network.

5.2.1. Description of the single-layer networks

The created distribution networks are simple single layer networks. The suppliers within the networks will consist of diesel generators (DG) generating $6600V$. The HUBs in the network will consist of high voltage switchboards (SWB) which will redirect the delivered $6600V$ charge to the users. These users will consist of transformers (TF) capable to transform the received $6600V$ to $440V$. Since only one distribution network is assessed, the $440V$ output of the transformers will be ignored. Figure 5.3 shows a few examples of the different networks that can be generated. To be able to give a simple description of a distribution layer, the following coding will be used to describe the single layer distribution networks:

$$(nr_{suppliers})S-(nr_{HUBs})H-(nr_{users})U$$

So, a simple single distribution network as shown in figure **??** consisting of two suppliers, two HUBs and two users will be; $2S-2H-2U$. Using this code, the following information can be generated:

- *Number of nodes*

$$nr_{nodes} = nr_{suppliers} + nr_{HUBs} + nr_{users} \tag{5.2}$$

- *Number of possible edges*

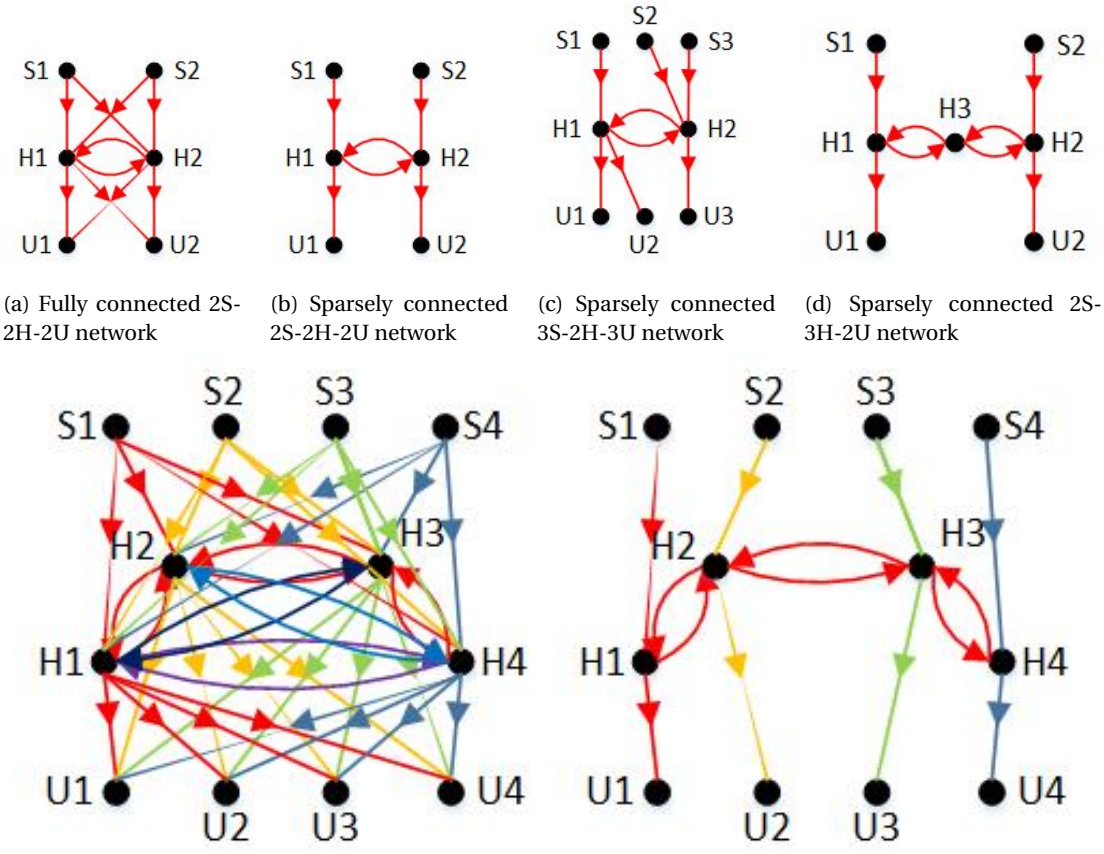
$$H \neq 0, \qquad nr_{edges} = nr_{suppliers} \times nr_{HUBs} + nr_{HUBs} \times (nr_{HUBs} - 1) + nr_{HUBs} \times nr_{users} \tag{5.3}$$

$$H = 0, \qquad nr_{edges} = nr_{suppliers} \times nr_{users} \tag{5.4}$$

- *Number of possible hits, n:*

$$n = nr_{nodes} + nr_{edges} \tag{5.5}$$

Using the described network code and the information that can be generated using this code, table 5.3 was created. This table states all the different networks that will be evaluated in section 5.2.2.



(a) Fully connected 2S-2H-2U network

(b) Sparsely connected 2S-2H-2U network

(c) Sparsely connected 3S-2H-3U network

(d) Sparsely connected 2S-3H-2U network

(e) Fully connected 4S-4H-4U network

(f) Sparsely connected 4S-4H-4U network

Figure 5.3: Example single layer distribution networks

The goal of the evaluation of the different networks shown in table 5.3 is to be able to describe how changing the composition of a distribution layer influences a distribution network. To keep all the networks equal with respect to the amount of resources that should be distributed, the total amount of resource input and output will be equal. The resource input and output is shown in table 5.3 as E_s and E_u, which is the maximum amount of resource given per supplier or used per user. Since the networks use a total number of users and suppliers of two, three or four, the total amount of resources to be transported is taken as 12 or $E_{tot} = 12$ ($E_{tot} = nr_{suppliers/users} \times E_{s/u}$).

CODE	nr_{nodes}	nr_{edges}	n	E_s	E_u
2S-0H-2U	4	4	8	6	6
3S-0H-3U	6	9	15	4	4
4S-0H-4U	8	16	24	3	3
2S-1H-2U	5	4	9	6	6
2S-2H-2U	6	10	16	6	6
2S-3H-2U	7	18	25	6	6
2S-4H-2U	8	28	36	6	6
3S-2H-2U	7	12	19	4	6
4S-2H-2U	8	14	22	3	6
3S-2H-3U	8	14	22	4	4
4S-2H-4U	10	18	28	3	3
3S-3H-3U	9	24	33	4	4
4S-4H-4U	12	44	56	3	3

Table 5.3: Different networks used for the single-layer evaluation

Finally, it will be necessary to provide some of the functional data regarding the networks. Since the cost of the network will be predicted using either the size of the network or the total length of the edges using some geometry, it is assumed that whenever more than one of each element (supplier, HUB, user) is present, it will be located into a different zone. This means, that in the case of $3S-2H-3U$ a total of three zones are present. These three zones each contain one supplier and one user. The HUBs are put in the earliest most middle zones. So, in this case the first HUB will be placed in the second zone, while the second HUB will be placed in the first zone. Furthermore, the transformers do not have a minimum required amount of input resource needed to function. Without a minimum required amount of resource per transformer a different importance factor for some of the transformers is also not needed.

5.2.2. Results

The different networks that have been discussed in the previous section will be used to determine how redundancy affects a single layer distribution network. The distribution networks used in this section are fully connected. The results generated can be used to enhance the understanding of the effect of redundancy on the distribution networks described in this research.

For the different networks, the amount of resources transported from the suppliers to the users has been assessed for different amount of hits. The results have been plotted for an increasing network size (both number of edges and total length of the edges). The results can be seen in figure 5.4 and 5.5 for fully developed networks.

Figure 5.4 shows the average capacity flow for single distribution systems with a certain number of suppliers (S), HUBs (H) and users (U). It can be seen that for a single hit, the average capacity flow increases as the network increases in size. But, there is an imbalance between the growth of the average flow and the size of the network. The 3S-3H-3U network has only half the size of the 4S-4H-4U network, while the average capacity flow is only 2,3% lower. Going a network lower, 25% of the size (number of connections) is lost with a decrease of 1% in the average flow. This imbalance constitutes to using a smaller number of nodes and edges to decrease the cost due to a decrease in size/length and amount of components needed. These trade-offs are the reason this method has been created. The situation of figure 5.4 changes when instead of a single hit, several hits are accounted for. This new situation is shown in figure 5.5.

The gradual increase in average capacity flow as the size increases as seen in figure 5.4 is gone once multiple hits are assessed. The four hits situation as shown gives insight into the shortcomings of some of the network combinations.

First off, the completely used four-zone network (4S-4H-4U) still has the largest average flow. The second and third largest network now have a larger difference with the largest network, but are still following closely with the average flow. The difference between the $4S-4H-4U$ and $3S-3H-3U$ networks was only 2,3% at 1 hit and has changed to 8,6% for 4 hits. Although the difference in average flow has increased, it should not be forgotten that the fully developed three-zone network has only half the size of the four-zone network.

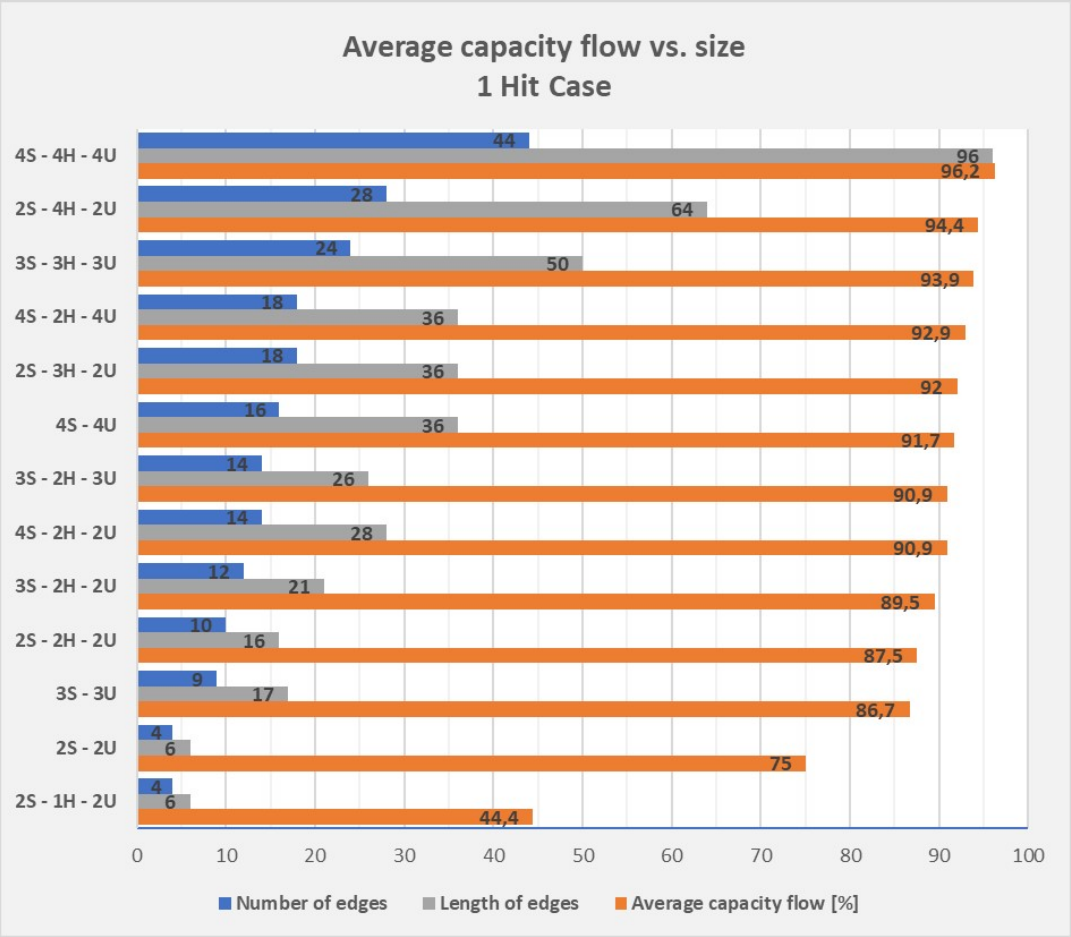


Figure 5.4: Average capacity flow vs. size for single distribution networks with a single hit

Secondly, there are three networks with a total length of 36 ($4S-2H-4U, 2S-3H-2U and 4S-4H$). These networks show that when purely looking at the length/size of a network it is the best option to only take suppliers and users and to leave the HUBs out[1]. However, leaving HUBs out is not an option as many smaller systems also need to be supplied which are not considered in the current research. The second strange thing these three networks show, is that a network with many suppliers and users that uses a low amount of HUBs is a worse scenario then no HUBs at all. While, a network with a small amount of suppliers and users, and a large amount of HUBs has a higher average flow. This can be explained due to the lower number of HUBs forming a choke-point in the network, creating extra opportunities to disable the network. A higher amount of HUBs creates a higher amount of reconfigurability within the network thus increasing the flow through the network.

Finally, the smaller networks ($size < 16$) although performing well for the one hit case, suffer greatly with multiple hits. This is to be expected, as they do not have the number of connections to support a network after several hits.

In conclusion, to generate a high amount of average capacity flow a large distribution layer is needed (both size and length). However, when concessions can be done on the average flow the size of distribution layer drastically decreases as both the edge length and the amount of components can be decreased. This trade-off between capacity and the size of a network give a chance to optimize a system topology to the needs of the user.

[1] Note: this case is based on the fully connected networks. When the networks are not fully connected, the networks without HUBs will have a larger increase in vulnerability then those that have HUBs.

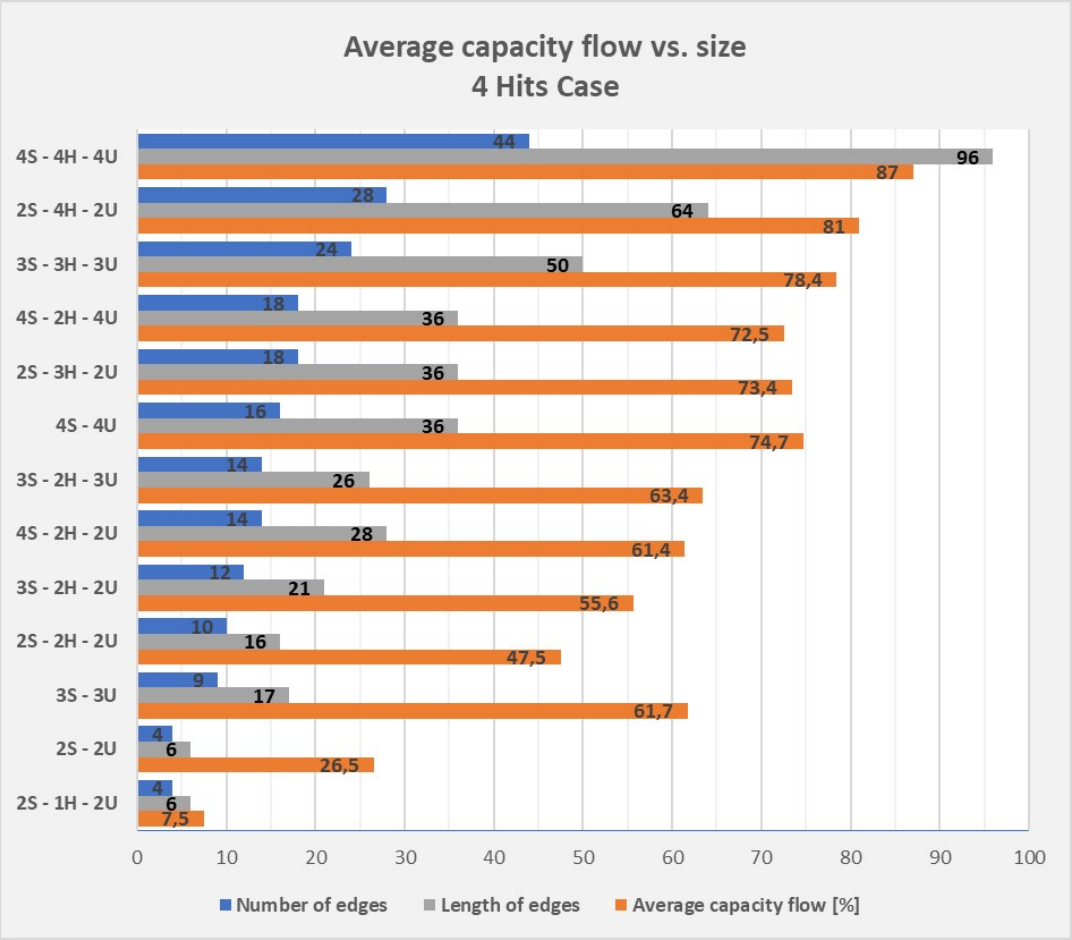


Figure 5.5: Average capacity flow vs. size for single distribution networks with four hits

5.3. Test-Case: A notional frigate

The previous results of the two-layer network have already shown that the method may be used to predict the vulnerability of system topologies. The current section is devoted to the application of the method on a notional frigate network. The results found in this section will be used to determine how the two vulnerabilities are connected to each other. This section will also show the results of a (manually designed) concept frigate network relative to the results of the method. This comparison between the concept frigate network and the generated network results will show the benefits of the developed method with respect to the current way of design.

5.3.1. Description of the network

The notional frigate network is based on the defined nodes by de Vos [2014]. These nodes can be seen in table 5.4. This table also shows the abbreviations (abbrev.) used in the graphical representation of the entire network (randomly connected) in figure 5.7, as well as the number of each node (Nr.) that is present within the system topology. Each of the nodes is subject to one or more distribution networks, this is shown in table 5.4 by the network type of each node. The seven distribution networks are also shown in their fully connected variant in figure 5.7. The other specifications shown in table 5.4 are component dependent [2]. Due to this dependency, all the different node types will have to be described. A note should be made about the zones shown in the table, these zones correspond with the following locations on the vessel:

1. *Aft*, the zone located in the back of the vessel
2. *Midship*, the zone located around the centre of the vessel

[2]Note: the capacities of the components have been altered and do not show the actual values used for the notional frigate

3. *Forward*, the zone located in the front of the vessel

The zoning of the vessel is also shown in figure 5.6.

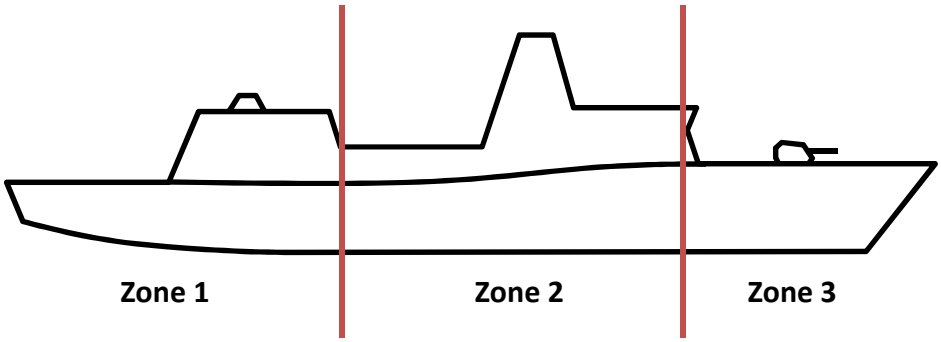


Figure 5.6: Zoning used for the notional frigate in the test-case

Table 5.4: Node specifications for the notional frigate

Node	Abbrev.	Nr.	Network type	Capacity [kW]	Minimum required input [%]	Zones
Diesel generator	DG	4	6600V	1500	-	[1 1 2 2]
High Voltage HUB	HV SWB	4	6600V	0	-	[1 1 2 2]
Transformer	TF	3	6600V	-3000	0	[1 2 3]
			440V	3000	-	
Low Voltage HUB	LV SWB	3	440V	0	-	[1 2 3]
Chilled water plant	CWP	3	440V	-450	0	[1 2 3]
			CWLT	1100	-	
LT Chilled water HUB	LT Pipe	3	CWLT	0	-	[1 2 3]
Heat exchanger	HE	5	CWLT	-1100	0	[1 1 2 2 3]
			CWHT	1100	-	
HT Chilled water HUB	HT Pipe	5	CWHT	0	-	[1 1 2 2 3]
Sensor 1	S1	1	440V	-750	70	[2]
			CWHT	-1100	70	
			DATA1	1	-	
Sensor 2	S2	1	440V	-150	70	[1]
			CWHT	-225	70	
			DATA2	1	-	
Computer room	PC	2	440V	-30	60	[1 2]
			CWHT	-45	60	
			DATA1	0	-	
			DATA2	0	-	
Command centre	CIC	1	440V	-75	60	[2]
			CWHT	-110	60	
			DATA1	-1	0	
			DATA2	-1	0	
			DATA3	1	-	
Weapon	W	1	440V	-500	65	[3]
			CWHT	-750	65	
			DATA3	-1	0	

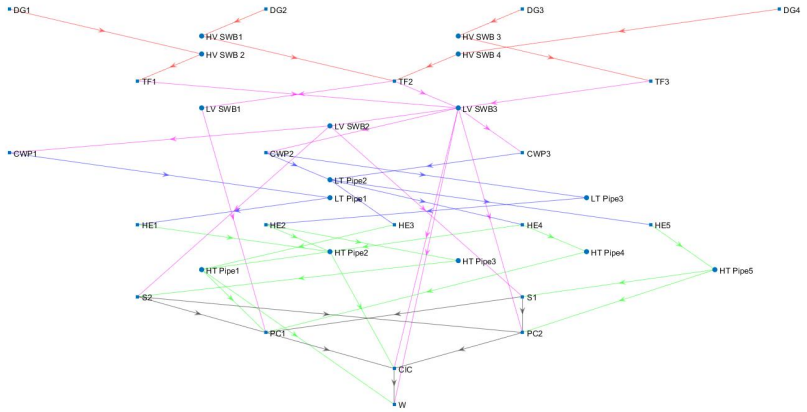


Figure 5.7: Random frigate graph [Note: The actual frigate network cannot be shown due to the classification of that information]

As shown above, the different node types will have to be discussed separately. These discussions are listed below for the different node types.

- *Diesel generator,*
  The diesel generators are used to generate a high voltage current that can be used by the transformers to generate a low voltage current. The diesel generators have been paired into two generator pairs. The first of these pairs is located in the aft zone. The second pair is in the midship zone. The diesel generators are the main suppliers of the frigate network, they have no input but produce an output. It was assumed that the diesel generators are always delivering power at their nominal power output of $1500kW$.

- *High voltage HUB,*
  The high voltage HUBs are switchboards that deliver power generated by the diesel generators to the transformers. The switchboards are located in the same areas as the diesel generators. This location is gained from the current frigate designs in which the line between generators and switchboards is minimized.

- *Transformer,*
  To change the $6600V$ supply to a $440V$ supply, transformers are needed. These transformers form the basis for the low voltage network and are separated over the entire vessel, every zone on the vessel contains a single transformer. The size of the transformer is taken to be twice the size of a diesel generator. The oversizing of the transformers ensures that (theoretically) a loss of one of the transformers should not lead to a decrease in capacity of the $440V$ network. This ensures a higher continuity for the most important power distribution system on board. Finally, the minimum input for the transformers is taken at zero, as they are able to transform the voltage level at almost any current level (at increasing losses for lower currents).

- *Low voltage HUB,*
  To low voltage HUBs or switchboards are used to distribute the $440V$ power over the users of the $440V$ distribution network. The HUBs are located in every zone since their suppliers are also located in every zone. Furthermore, the different system that need to be supplied are spread out over the entire ship.

- *Chilled water plant,*
  The chilled water plants are used to create the first stage of the chilled water network. They produce low temperature (LT) chilled water at around $1,5kW$ chilled water per kW $440V$ received. They are assumed to have no minimum requirement to the amount of electricity needed to function. To spread

out the suppliers of chilled water, the three chilled water plants have each been placed into a different zone.

- *LT chilled water HUB,*
  The chilled water HUBs or pipes are a complex network of pipes that need to be taken as a single distribution point. For the sake of this research, it is assumed that the distribution of chilled water is achieved in a single localized HUB node. These HUBs have been spread out over the different zones as the piping is also present in each zone of the vessel.

- *Heat exchanger,*
  The function of the heat exchanger is to use the received LT water input to create an outgoing flow of High temperature (HT) water. In total five heat exchangers have been placed in the frigate with a capacity each of $1100kW$. Table 5.4 shows that each heat exchanger has the same input requirement as the output of each chilled water plant. Thus, two heat exchangers could be destroyed without a loss in capability. This overcapacity has been taken from the frigate design and is based on the different load characteristics for different systems and the different modes in which the vessel can operate. The minimum requirement for the heat exchanger has been assumed zero.

- *HT chilled water HUB,*
  The HT chilled water HUBs or pipes are the same complex networks as the LT HUBs reduced to single points. In this layer five HUBs are present since five heat exchangers are present.

- *Sensor 1 and sensor 2,*
  The first sensor is placed midship and is the larger of the two sensors on board, while the second sensor is smaller and located at the aft of the vessel. From the electrical load balance of a frigate, the electrical input has been determined as well as the minimum requirement to be operated. The chilled water requirements have been assumed to be 1,5x the electrical input. This amount of chilled water is used to chilled the heat produced by the electrical power and the heat taken from the surroundings. The minimum requirement of the chilled water input is taken equal to the electrical requirement because of the scaling of chilled water as assumed before. The final distribution network that sensor 1 is a part of is the $DATA1$ network which is the data generated by this sensor. The value of the data is taken as one as a value is needed, but if the sensor is working enough data will be generated to be processed. The output of sensor 2 is $DATA2$ and follows the same assumptions as shown above.

- *Computer room,*
  The two computer rooms of the frigate function as data HUBs for the two sensors and their data heading towards the command center. The electrical load and minimum required electrical load are both found from the electrical load balance of a frigate. For the chilled water, the same assumptions have been made as for the sensors. Each one of the computer rooms is in a zone with a sensor.

- *Command center,*
  The command center is used to process all the sensor data and to create data that can be sent to the weapon. It is thus that the command center has two data inputs; $DATA1$ and $DATA2$, and a data output; $DATA3$. The electrical and chilled water load and its requirements have been based on the same assumptions as for the computer rooms and sensors. The command room is located midship.

- *Weapon,*
  The final component of the frigate network is a weapon. The requirements of this weapon have been gotten from the electrical load balance and the assumptions as shown by the sensors. The final input is $DATA3$ used to drive the targeting of the weapon. The weapon system is located forward.

The above components form a framework for which a vulnerability prediction should be made. The final step to create a problem for which solutions can be found is to determine which node(s) is/are the prime user(s) and its their importance for the network.

In this case, the weapon system is taken as the prime users. This weapon system is given an importance factor (IF) of two. Using the importance factor, the nodes that have no real benefit to the entire network will be ignored where possible. The state of the system topology is simple for this case with a single prime user. The state can be described as:

$$state_{network} = \begin{Bmatrix} 0, & if\ the\ weapon\ works \\ 1, & if\ the\ weapon\ fails \end{Bmatrix}. \tag{5.6}$$

5.3.2. Results

The networks generated by the SDS-ATG tool for the common frigate using the data from the previous section are evaluated by the vulnerability assessment method. The results of this evaluation can be seen in figure 5.8a and 5.8b. These two figures show the vulnerability of both parts against the number of connections present within the networks. The figures consist of data from four different hit simulations namely: one, two, three and four hits[3].

The first thing that can be seen from the two figures is that there is an increase in the vulnerability when the amount of hits increases for the topologies. This increase in the vulnerability is logical, as a network with a fixed amount of connections will have more chances to fail as more parts are hit at once.

The second effect worth noticing is that the pareto front for higher number of connections straightens into a constant vulnerability line. For the connectivity part, this happens at around 80 connections. The capacity part has this point at around 120 connections. This line shows that at a certain point, no or almost no improvements can be gained with respect to the vulnerability by increasing the networks size. In other words, a maximum size for the network exists with respect to the vulnerability of the network. This effect also exist on the other edge of the network size. Since, a minimum possible number of connections can also be seen on the left side of figures.

Thirdly, it can be seen that not only does a maximum size for the topologies exist with respect to the vulnerability, but for this these larger size, the vulnerability of part-I and part-II is almost equal. This effect of the score of both parts being equal enters at around 110 connections. The topologies with more connections are not influenced by the resource requirements as the hits to the critical components have already been handled by the connectivity part.

Finally, the vulnerability increases when using part-II on the results of part-I. This fourth behaviour of the results is to be expected, as networks that are still connected can have the problem of not being able to supply the right amount of resources to the prime users. However, the increase in vulnerability is remarkable, as many topologies reach a 100% vulnerability in part-II of the method.

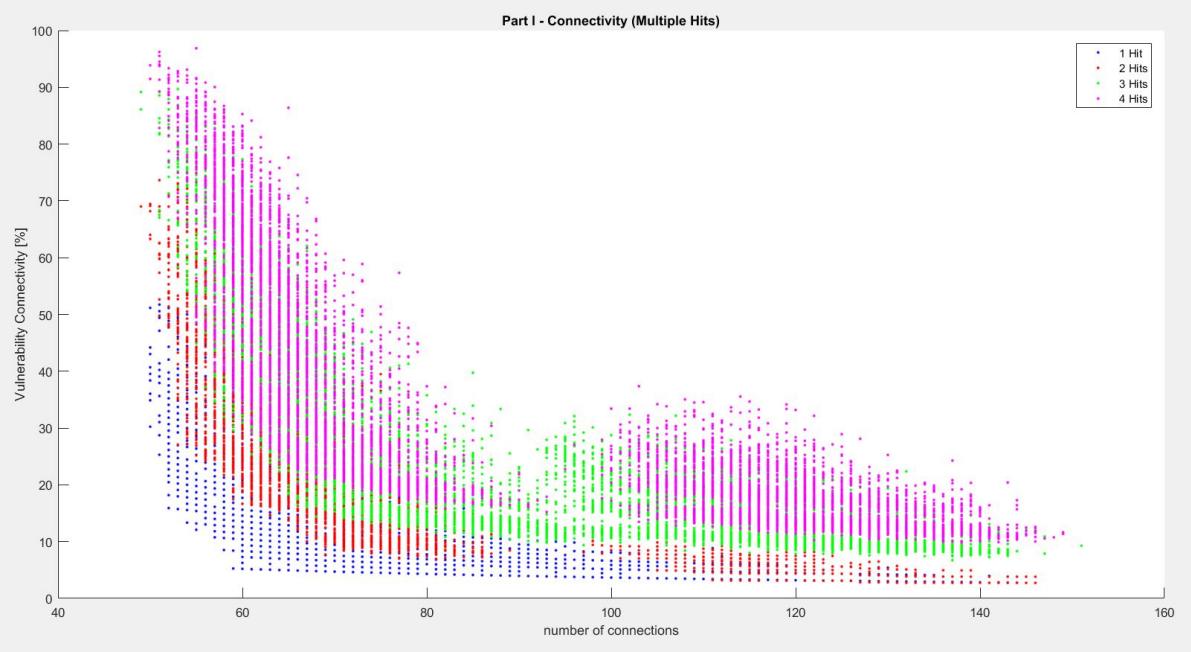
A small amount of the extra failures can be explained. The HUBs are ignored when looking for the connections between nodes, thus a HUB that has an input requirement (e.g. the server which requires cooling and electricity) and which is not a critical user will not be checked [4]. A hit to the input of this HUB will lead to failure in the capacity part. For the other topologies, separation must have occurred within the network leading to a situation in which some of the energy requirements cannot be fulfilled. However, these figures do not show how the different networks are correlated to each other.

To show the correlation between part-I and part-II of the method, figure 5.9 shows the pareto front of all of the four hit scenarios. This front is shown in red and depicts part-I connectivity. The magenta line shows the result of part-II capacity for the topologies within the pareto front. The four different pareto fronts show that their is a great gap between the two scores for the vulnerability. However, although this gap is great, it does show a correlation between the two fronts. For the one hit case, the magenta line follows the pareto front for the larger networks, but when the network decreases in size jumps in vulnerability appear due to the network still being connected but not being able to supply the right amount of energy. This effect is still present in the multiple hits scenarios, as they also show a disjoint between the scores of both parts. To show the exact correlation between the two scores, figure 5.10 was plotted.

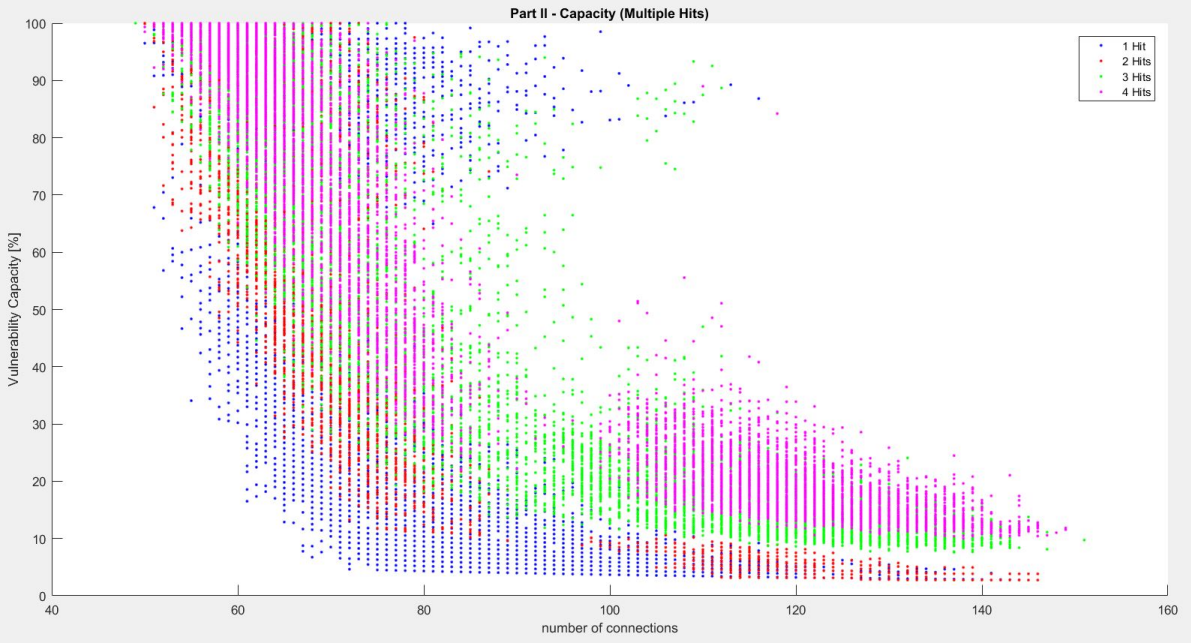
Figure 5.10 shows the values of the connectivity score opposed by their capacity score for all of the topologies with a single hit. This graph shows that for networks with less connections, the chance of a large difference between the two scores is the greatest. When more and more connections are added, the scores are growing closer to each other (bottom left corner). This result is as expected, as the chance that a smaller network missing some connections is not able to fulfil its requirement is larger than that of a network having more connections and not having much trouble with losing a single connection. However, there are excep-

[3] Although only four hit simulations were used, a bug in Matlab demanded that the longer simulations (two, three and four hits) were separated in two parts. These two parts consist of a lower part with a number of connections lower than a hundred and an upper part with more than a hundred connections. This can be seen in both figure 5.8a and figure 5.8b, as a part of the topologies is missing just before the hundred connections mark.

[4] This simplification was discovered at to late an stage to implement a solution, thus the solution is given in the recommendations in section 6.4.



(a) Results part-I connectivity



(b) Results part-II capacity

Figure 5.8: Results generated by the vulnerability method

tions to this, as it can be seen that even for the larger networks there are possibilities for which the capacity vulnerability is much larger than the connectivity vulnerability (upper left corner). These networks must have their connections packed together creating weaknesses for the entire network at other points.

In conclusion the connectivity score gives a reasonable estimation although it can be wrong. However, when a more accurate estimation of the vulnerability is necessary part-II provides this.

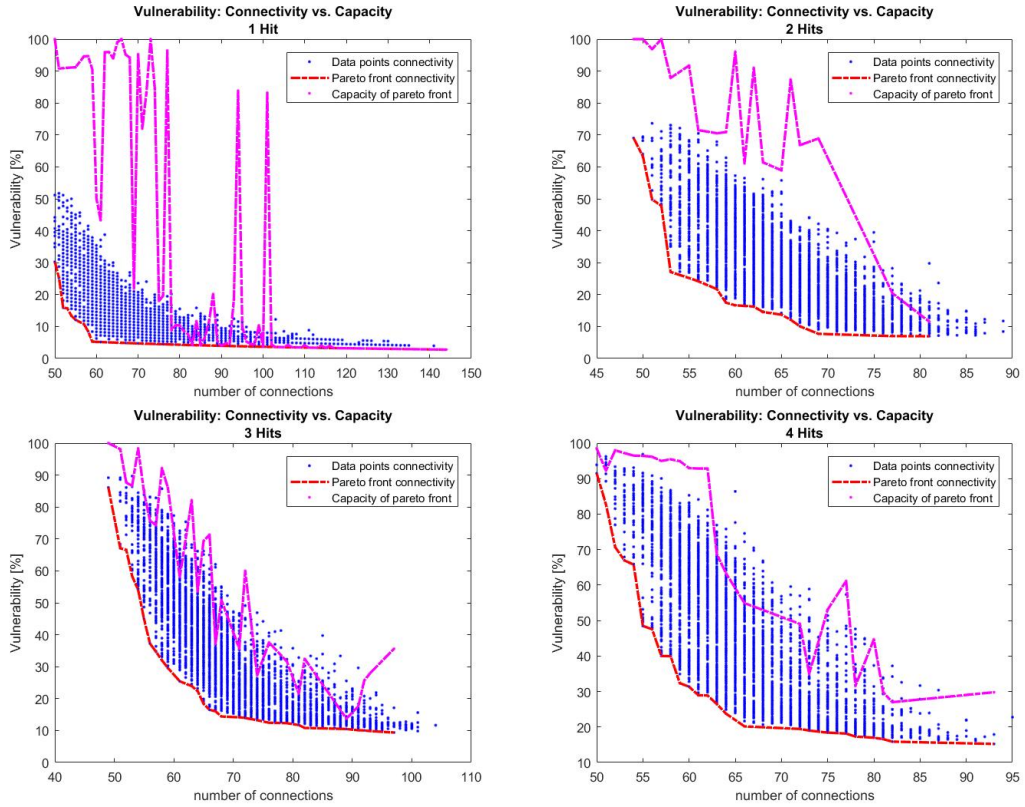


Figure 5.9: Difference between the connectivity and capacity results shown on the pareto front

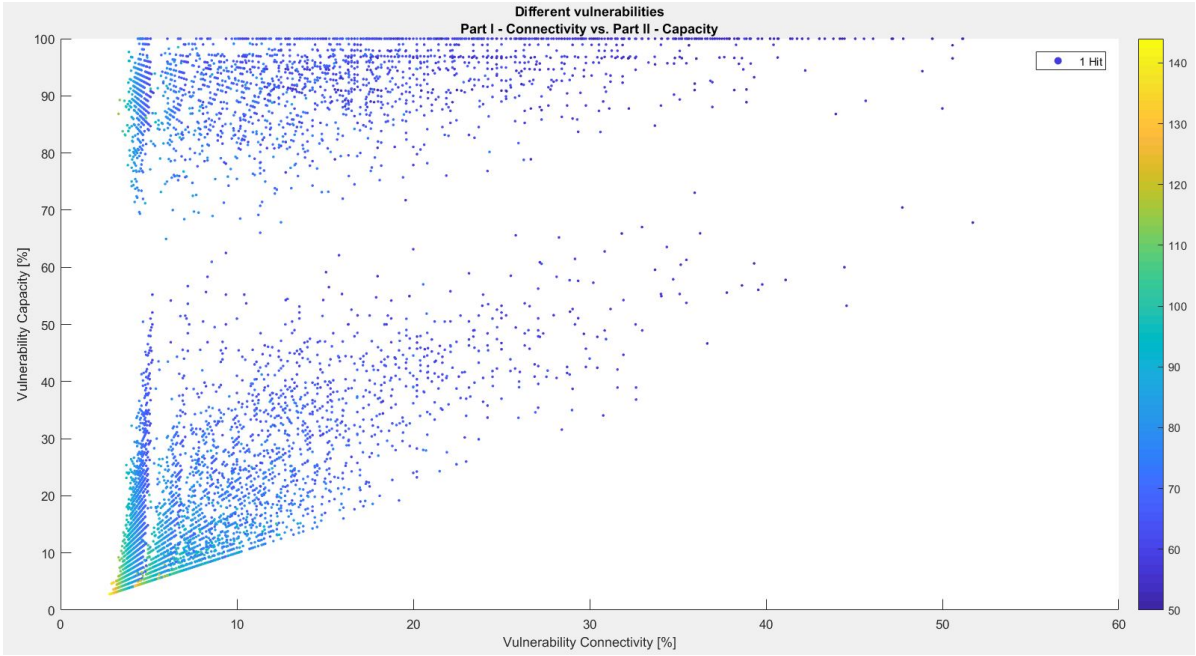


Figure 5.10: Results of the tool for the connectivity part of the method

5.3.3. A real concept versus generated results

The results in the previous section show what can be done with the tool when it is used to determine the vulnerability of randomly generated system topologies based on the components of a notional frigate. This section will show how a real frigate concept based on those same components fits within data generated by the SDS-ATG tool.

Although the frigate concept uses the same components as described in section 5.3.1, the required amount of resources by the components is different. The total amount of required resources is equal, but some parts of the system have been optimized in the concept. Due to this difference, new data has been generated with the capacities of the concept frigate to keep a comparison viable. The frigate has a topology setup that is similar to the bare minimum required for a topology to still function according to the SDS-ATG. The HUBs are mostly interlinked to each other while each converter is only connected to a single HUB (with an exception within the end users). The vulnerability data for the frigate can be seen in table 5.5 and figure 5.11 and 5.12 show how this corresponds with the generated data.

HITS	PART I – CONNECTIVITY VULNERABILITY [%]	PART II – CAPACITY VULNERABILITY [%]
1	22,3	47,6
2	39,4	71,1
3	57,4	86,5
4	67,2	91,9

Table 5.5: Results of the real frigate concept network

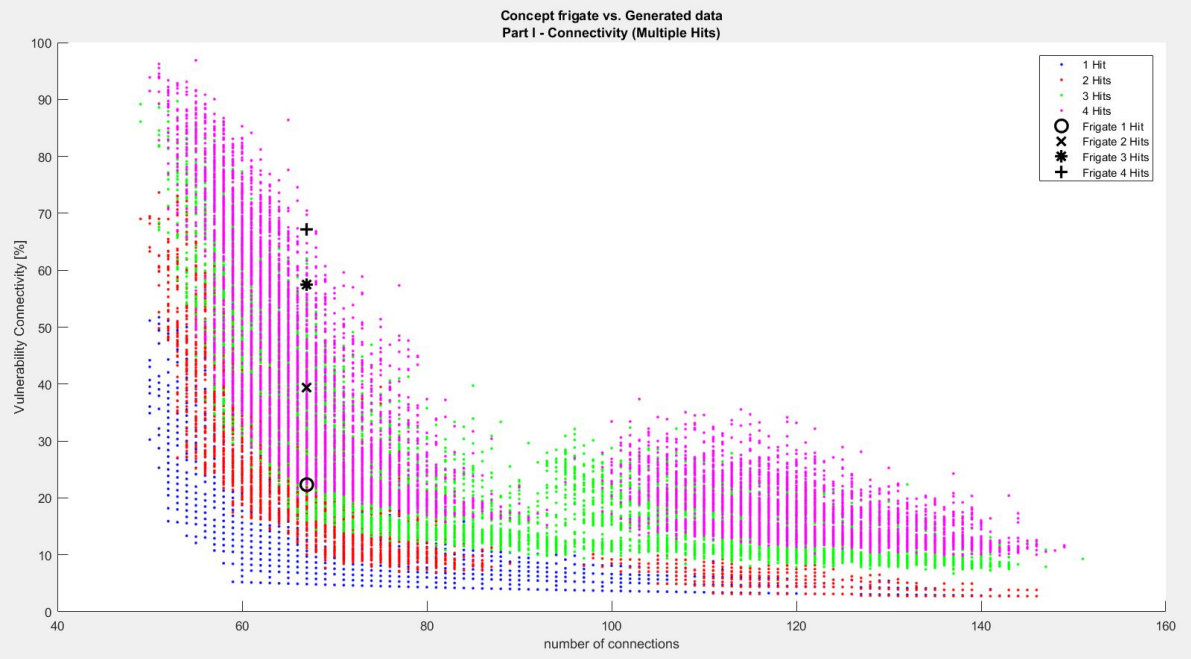


Figure 5.11: Results of a concept frigate within the results of the method part-I

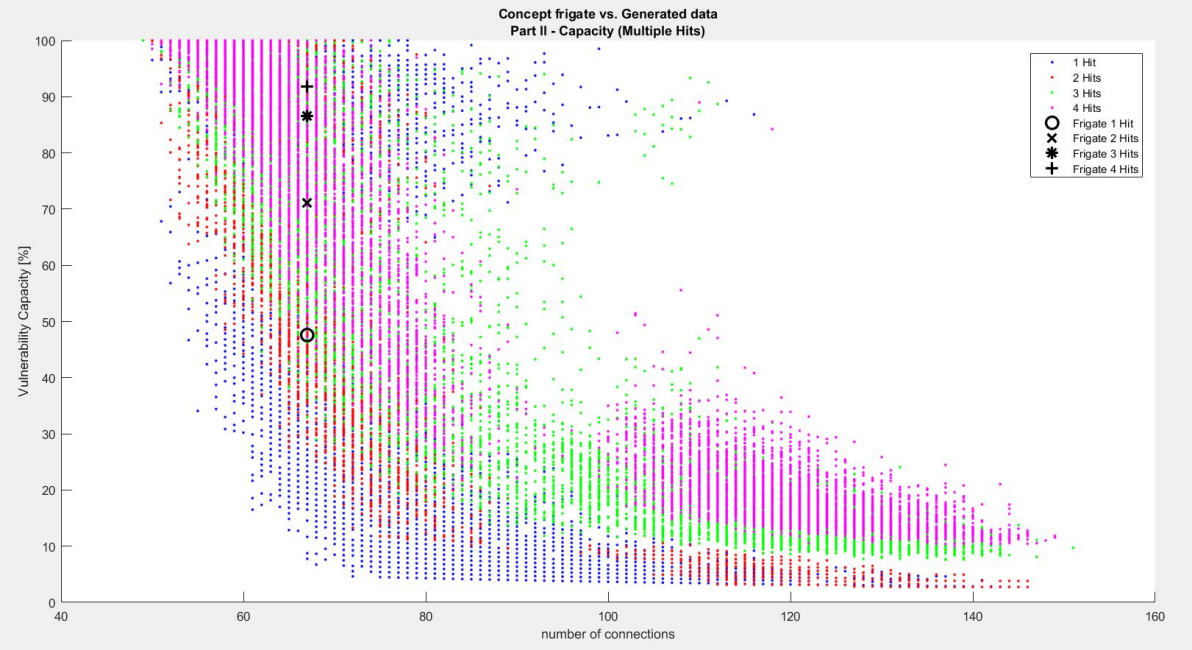


Figure 5.12: Results of a concept frigate within the results of the method part-II

The above two figures show the generated data with the frigate data added. The figures show that the frigate concept is behind the pareto front of the generated data for the different hit cases. With the concept shown not to be an optimal solution, a solution on the pareto front, an improvement can be made within the current designs with respect to the system topology. However, there are some side notes for the evaluation of the concept design. These notes are:

1. The network used to evaluate the design has been generated using technical data. The used data has a high amount of detail, but at some points there can be discussion about the right identification of HUBs and connections. For instance, the pipes of the chilled water network are modelled as HUBs although nowhere on board is an actual HUB. The chilled water network consists of pipes with offshoots to systems. Thus, the model does not completely reflect the reality influencing the vulnerability found.

2. The generation of the topologies as done by the SDS-ATG tool only accepts a single connection between two components. From the concept data it can be seen that at some points multiple connections are established between two nodes. For instance between one of the sensors and a low voltage HUB. This simplification is assumed to be harmless, as said connections are close together. However, using multiple connections can have a positive impact on the concept vulnerability, but it would also increase the total number of connections thus shifting the pareto front to a more positive vulnerability.

With the side notes shown above, it can still be concluded that an improvement can be made with respect to the concept design. This improvement is mostly based on a more redundant design, a design with either more HUBs connected or prime users connected to several HUBs. These improvements are based on the earlier mention of the topology having a small amount of connections. This small amount of connections influences some of the distribution layers, as some of these layers have seperate networks. This seperation of networks creates a higher vulnerability as a smaller number of hits is able to disable parts of the system.

Finally, it should be added that some of the problems met with the current vulnerability assessment as mentioned above can be mitigated by routing the topologies. By given the topologies a more spatial design, redundant connections between two components can be added as well as an increase probability of a hit for longer connections. This will cost more time, but will generate a more accurate vulnerability assessment.

5.4. Conclusion

The first section of this chapter has shown that verification of the developed method for multi-layered networks. With the results in this first section, the next step is to determine how the results can be interpreted for

different networks. This first interpretation of the networks is done in section 5.2, in which different smaller layers are compared to each other.

The comparison from section 5.2 is shown in the results in section 5.2.2. The comparison shows that the average flow through the network increases as the number of connections within the network increases for a single hit case. However, there is an imbalance between the growths, with the larger networks having an increase of 25% of their connections while the average flow only increases with 1%. A trade-off can be made between these two objective scores to find a balanced solution. When adding more hits, as in the four hits case, the same behaviour can be seen with bigger differences for different network compositions. Of these compositions, the networks without HUBs have a rather high average flow compared to their number of connections. The only problem here being that in these cases a complete star network is generated. For the networks with HUBs, it can be seen that a network with more HUBs than suppliers/users behaves in a more positive fashion. For instance, figure 5.3 shows that for 36 connections, a combination with an extra HUB has a higher average flow than one with extra suppliers/users.

The last results sections shows the test-case in which a notional frigate is used to show the results of a system topology. The first thing that is seen, is that the vulnerability of a system increases as the number of hits increases. Secondly, the topologies evaluated with part-I are evaluated with part-II. The capacity check shows that the vulnerability increases for smaller topologies (lower number of connections), the larger topologies have a vulnerability close to their original. This effect is also plotted in figure 5.9 were the objective score of part-I is matched to part-II. Finally, a concept is used to show how a system from the current way of design fits within the generated data. This final plot shows that improvements can be made with respect to the topological layout of the system. However, without routing the topologies through a vessel, it will be impossible to say for certain that much improvements can be made.

6

Conclusions

The final chapter discusses the conclusions that are obtained from the research that has been done. First, a short summary is given of the problem and the proposed solution approach. The application of the found solution will be discussed next, followed by the conclusions of this research. Finally, a few recommendations are given based on the experience of this research.

6.1. Summary

The focus of this MSc thesis was to determine a new vulnerability assessment method, based on network theory, that can be used to assess the vulnerability of distributed system topologies. These system topologies are created by the tool SDS-ATG currently in development as part of the PhD research of de Vos.

Literature on network theory provided a large amount of existing methods to determine the 'vulnerability' of a general network. These methods were used to determine whether a useful assessment already existed or not. Without finding a usable method, lessons were learned with respect to the applicability of certain vulnerability aspects, these are:

1. A generalization of the network topologies cannot be used, as the system logic does not allow it.
2. Every node should be evaluated individually, because of the dependencies within the system.
3. Deterministic methods should be used.

In most cases the methods frequently used to evaluate networks were incapable to handle the system logic present, as mentioned above, within the system topologies. The assessment tool present within SDS-ATG used parts of the system logic to determine the 'best' topologies, however the outcomes of this tool had a topology that was not desirable as it needs a standard user-override to generate redundant connections.

With the lessons learned from the currently available methods, a new method was created that uses two different approaches to determine the vulnerability of a network. Both approaches use a set of damaged forms of the network under assessment to determine its response to such damage states. The first part is based on the connectivity of a network. It is determined whether the prime users of the network are still connected to all their sources and could thus function. However, all of the sources should also be connected and thus a reversed search tree is created to find if prime users could receive the resources needed to function. The second part uses the capacity of the components within the network to determine the vulnerability. Based on the outcome of the first part it is known whether the prime users of a network could receive their resources. This part adds to the first part by determining if the prime users are receiving their resources. It uses the capacity of the different components to determine if enough resource flows are available to let the entire network function.

With the outcome of the vulnerability assessment method and the size of the network available, a genetic algorithm is used to optimise the topologies for both vulnerability of the system and its size. These solutions shown in a pareto front show the networks for which both the vulnerability and the size cannot decrease without increasing the other. This data could then be used to determine the best-balanced design solution with respect to the size of the network and its corresponding vulnerability. The outcome should create a better decision making with regard to this design trade-off.

To verify the outcomes generated by the assessment method, hand calculations were used for small networks that include every situation possible with SDS-ATG. The assessment method was then used to determine the vulnerability of a concept design to show the improvements that can be made on current designs using the method.

6.2. Application

The vulnerability assessment method determines the vulnerability of a system topology. The method can find the vulnerability of a system design for several different states, with states being the functioning of designated prime users. These states govern the functions that the entire system design is still capable to execute. The different states together with the ability to determine the size and the level of detail of the topologies created by SDS-ATG gives the tool (combination of SDS-ATG and assessment method) a wide variety of applications.

1. *Design tool (early stage design),*
   To find a balanced design solution during the early stage design. The failure rate, either by internal or external factors, of early concepts can be explored using SDS-ATG and part I of the method. The first part only needs the components and their respective distribution networks to function, but it already gives an indication of the vulnerability that can be expected of the system. At this stage of the design, the addition or subtraction of components, for instance to create redundant systems, can be explored to see the change in vulnerability of the system of interest.

2. *Design tool (detailed design),*
   After the early stage design, the method can be used when more information is available to the engineer. Using the capacity of the components a more reliable system vulnerability can be found using part II of the method. It is at this point that a broad exploration of all the topologies of the system can be done with the current set of components. This exploration will lead to insight of combination of connections that will or will not work. This is also the point where routing could be introduced to get an even more reliable assessment.

3. *Finding weaknesses,*
   Using the different states that can be implemented into the method, the different systems that fail during the assessment that lead to total failure can be isolated. This gives the possibility to find the weak spots of a system topology, as the following is known: which components fail, why did these components fail (which edges/nodes where hit) and how many times these components have failed. Using the failure data, the current system design could be altered to increase its vulnerability.

4. *Assessment of current designs,*
   Assessing current system designs can show why certain design decisions that have been made are good/bad. It will also show were the problems are, using the failure data. So, improvements can be made during possible re-outfitting of the vessel in the form of mid-life upgrades.

5. *System sizing,*
   Using the method to assess a system, one could find that certain components have been oversized or are to small. This conclusions can be drawn from the data with respect to reasons of failure. For instance, if the frigate network is taken. When the assessment concludes that failure is never caused by the diesel generators, the engineer could downsize the generators to reduce costs until a point is reached were the generators are the problem. But, just before this point costs can be saved while the system topology keeps the same reliability.

6. *Adaptations*
   With the adaptation of the method for different tools beside SDS-ATG, it would be possible to estimate the vulnerability for the routed paths of distributed networks. Since, the only difference with routing is that the edges are changed into components with locations.

6.3. Conclusions

The goal of this thesis was to show how the vulnerability of a conceptual design for distribution systems generated by SDS-ATG could be determined. This section will discuss the conclusions that have been established during this thesis. All of the research questions will be answered here to create a satisfactory answer to the main research question of this thesis.

RQ1 *Which problems result from using network theory to describe system topologies, and how could these problems be solved?*
Two problems were identified in section 3.4.4. First, the identification of the nodes as either a supplier/HUB/user isolates a group of components that have different roles at different times (e.g. battery, flywheel, etc.). From storing energy at one moment, to discharging energy at another, these components will have to be dealt with by creating two separate cases. Each case represents a different role of the component, such that a battery would have two states (e.g. storing and discharging energy). The second problem found is due to the determination of the connections between the components using network theory. The connections can either exist or they do not exist (binary). This negates the possibility of having redundant connections between two components, as a connection between components is always seen as the same thing regardless of the number of connections between those components.

RQ2 *Which generic methods and metrics exist to assess the vulnerability of networks in network theory?*
In section 4.2 different metrics from the network theory have been evaluated for their usefulness of evaluating the vulnerability of system topologies using the topological characteristics of those topologies. These evaluations concluded that not every metric can be used as an objective function within the NSGA-II algorithm. Furthermore, the metrics that could be used had trouble dealing with the system logic which led to a distorted outcome with more vulnerable topologies being rated higher. This can also be seen in section 4.2.4. Thus, estimating the vulnerability is better than predicting the vulnerability based on topological characteristics.

RQ3 *What is missing from the custom evaluation method of the SDS-ATG tool?*
The custom evaluation method or the maximum HUB flow method, is discussed in section 4.3. The method utilizes the maximum flow available in the HUBs to make a prediction about the vulnerability of the system topology. As shown in section 4.3.3, the maximum HUB flow method has its flaws. First off, the method assumes that every supplier/user is connected to the HUBs with only a single connection. When a critical user needs more connections, these have to be hard-wired by the designer. Secondly, the importance of certain distribution networks is not taken into account, all HUB layers have the same importance. Finally, it is a method using the topological characteristics in the same manner as the metrics found in literature, but with the use of some system logic. The method does not take into account damage and the effects it has on a topology.

RQ4 *How is vulnerability defined in the case of distribution systems on board vessels?*
Based on the sub-research questions, the vulnerability has been defined as the chance that the topology does not meet the requirements given. These requirements have been determined for two levels of detail. The first level of detail uses the availability of connections from the prime users to their suppliers as a criteria. When the prime users are connected, the second level determines whether they receive enough resources to function. The requirements need to be met for modelled failure(s) within the topology.

RQ4.1 *How should vulnerability be interpreted?*
The vulnerability of the distribution systems is defined by observing the consequences of failure (e.g., the loss of one or more components or connections due to internal or external factors) within a system topology, see section 4.4. Both parts of the vulnerability function interpret the vulnerability in their own manner. For the connectivity part, the vulnerability of a topology is the chance of the prime users not being connected to all their suppliers (disconnected network), as stated in section 4.4.3. The second part of the tool uses the vulnerability as the the chance of the prime users not being supplied, see section 4.4.4.

RQ4.2 *At what level of detail should the vulnerability be determined?*
The previous sub-research question shows that the level of detail used in the vulnerability function differs for each of the two parts. Part-I connectivity uses a simple existence of a connection between the prime users and their suppliers to generate a first vulnerability estimate. The simple connection check can be easily used and gives a good indication of the topology vulnerability. The second part uses a higher level of detail. In part-II the resources available to the prime users are identified and used for the vulnerability. These resources are taken to be power [W] as a further separation of effort and flow would introduce time dependant aspects to the function, see section 4.4.4.

RQ4.3 *Should damage be inflicted to the network to predict the vulnerability?*
To gain insight in the consequences due to failure within the topologies, modelling the damage done to the topologies is necessary, see section 4.3.3. The failure that occurs within the topologies will be modelled as random failures due to the randomness of damage/failure on board a real ship, as shown in section 2.3.

MRQ *How to assess the vulnerability of a conceptual design for distribution systems generated by SDS-ATG (which is based on network theory)?*
With all of the research questions answered, a solution has been found to the question shown above. The vulnerability of a conceptual design for distribution systems generated by SDS-ATG can be assessed using the connections to the prime users and the amount of resources available to these users, as shown in section 5.1. The case study that followed this verification in section 5.3, showed that the connectivity check gives a reasonable estimation in an earlier design stage. For a more accurate estimation, as seen in figure 5.10, part-II capacity should be used as more detail is taken into account. Finally, the test-case with a concept of a notional frigate as seen in section 5.3.3 proved that improvements can be made in the current way of designing the system topology, as an improvement for the tested concept can be gotten for all of the scenarios shown.

6.4. Recommendations

Following the research done during this MSc thesis, some limitation of the designed method have been found. These limitations have led to several recommendations to improve the method and for further research.

- The first recommendation is regarding the components that cannot be clearly identified as either a supplier/HUB/user, see section 3.4.4. The role that components can take within a network is currently fixed within the SDS-ATG tool. This means that some components (e.g. batteries, flywheels, etc.) cannot be identified. For these components, different scenarios should be defined. For instance, if a battery is taken two scenarios can be identified. One state is of the battery charging (user) and the other is of the battery delivering power (supplier). In most cases however, some of the scenarios are less meaningful. In case of the battery, the scenario were it is a user could be left out since this will mostly occur during non-critical situations.

- Redundant connections are currently not allowed, see section 3.4.4. However, adding this kind of connection, for instance: two redundant power lines between two switchboards, would increase the chance of the HUBs staying connected and thus decrease the vulnerability of the system topology, see section 5.3.3. And since this is a possibility in real networks, it should be implemented.

- Dependant components, the current way of estimating the vulnerability cannot handle the dependency between two components, see section 4.4.5. If this problem is solved, it would give rise to a far larger group on network topologies that could be assessed.

- The chance of failure of components and connections is currently taken to be uniform, see section 4.4.5. However, it should be clear that not every system on board of the vessel has the same chance to fail (i.e. a system below the water line would not as easily be damaged by a surface hit on a naval vessel. In commercial vessels, the chance of a component failing is different for each component.) Introducing a hit chance for every component and/or edge would lead to a better accuracy of the vulnerability estimation. This could be accomplished by combining the work with reliability estimates of components (internal failure).

- Part-I connectivity, currently the HUBs in the connectivity check are skipped if they are not added as a critical user, see section 5.3.2. This feature was added to decrease the time needed to check a topology for the connectivity vulnerability. However, when a network is examined that has HUBs that need to be supplied with a resource to work, the skipping of HUBs becomes a problem. A solution to this problem is to use a connectivity check in which every node is checked independent of their job within a distribution layer. The time needed to run the check will increase, but a HUB that is also a user/supplier will be considered without any problems.

- Routing was mentioned in section 5.3.3 to mitigate some of the problems met during the network assessment. With routing, the topology is routed through a 3D model of the vessel resulting in a more

favourable failure characteristic of the network with respect to both internal and external forces. Also, connections are enlarged to span several compartments, which are different for every connection, instead of having an equal length. This real length of the connections gives a more reliable size estimate of the network.

The above recommendations followed from the research and the conclusions that can be drawn from it. Besides these, a few more recommendations can be added with respect to an increase of accuracy of the method and the overall evaluation of the network topologies. These recommendations all have to do with adding extra information, some of the information is only available in later design stages, but some can be added right away.

- Location, currently the location of components is based on either the zonal distribution or is not determined at all. Increasing the amount of information on the location of components could lead to a better estimation of the length of the edges and thus the length of the connections needed between components. It also leads to a generated topology that is more oriented to finding fitting connections between components, with the shorter connections between two layers being the best decision. The overall system topology would be better applicable as the placement of components would have been taken into account during the generation of the topologies.
- Edge capacity, introducing the edge capacity could influence the flow throughout the system during the capacity part of the method. Currently the edge capacity is set at infinite due to the lack of defined constraints. However, in some cases the flow through a connection cannot be infinite (i.e. a pipe cannot have an infinite flow or a cable cannot have an infinite current, as the voltage is fixed).
- Cost, since the size of the network is currently used to determine the cost an increase in accuracy in this objective can be provided by entering the cost as a function of each component/connection. For the current way of measuring size against another, the cost of the different edges would penalize topologies that have fully developed high value connections giving a more accurate approach to the cost of the network. If also the cost of the components is introduced it would be possible to compare different topologies that consist of different components with each other.

Appendix A: Determining the Sample Size

This appendix is concerned with the determination of the sample size. It will show how the sample size should be determined as well as the proof that the used method to determine said sample size can be used.

The sample size is important to reduce the amount of time needed to calculate an answer. To show that this reduction is important, the following example is given.

Example: A network with a total of hundred nodes and components is used to calculate whether a network is working or not. The time needed to calculate each possible outcome takes one second. Furthermore, the amount of possible hits within the network is increased. This example can also be seen in figure A.1.

Table A.1: Calculation time for a network based on the possibilities within the network

Number of nodes and edges = 100 Time to calculate one possibility = 1 [s]		
Number of Hits	**Number of possibilities**	**Time**
1	100	100 [s] = 1,67 [m]
2	9.900	165 [m] = 2,75 [h]
3	970.200	269,5 [h] = 11,23 [d]
4	94.109.400	1089,23 [d] = 155,6 [w]

From the above example, it can be seen that as the amount of possible outcomes increase, the amount of time needed does as well. To reduce the time needed to find a solution to the vulnerability of a network, it is necessary to determine a certain sample size that can give a reasonable accurate solution. To find the correct sample size estimation, it is necessary to identify the probability problem that is described by the method.

The number of possibilities n that are available in the example of table A.1 is the population corresponding to the number of hits to the network topology. The overall vulnerability is based on the behaviour of each individual within this population. In other words, each individual is a discrete random variable X with $X = [0\ 1]$, with a chance $P(X) = 0,5$. Thus, the probability density function is governed by a binomial distribution, this distribution is shown for several n in figure A.1. This figure shows, that for $n = 512$ the binomial distribution can be describe by the normal distribution (red line). Thus, when n increases, the binomial distribution takes the form of the normal distribution. From this point onward, it is assumed that the normal distribution can be used to describe the probability density function of the networks for large n. Using the normal distribution, it is possible to determine the sample size based on the standard normal table or Z-table. A part of this table is shown in table A.2, the entire table can be found in appendix .... This table is used to determine the Z-value of a given confidence level. The confidence level shows the certainty that the average vulnerability found from the sample size is within bounds of the confidence interval. This interval shows the bounds around the 'real' average vulnerability.

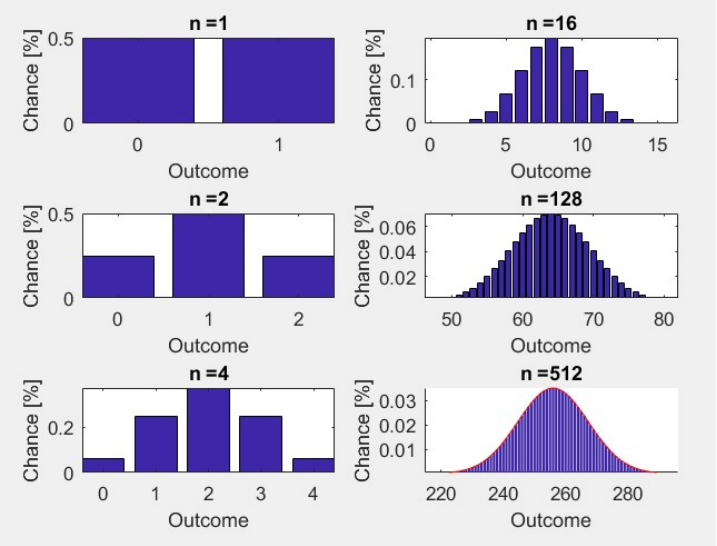


Figure A.1: Binomial distribution for several values of n

Table A.2: Part of the normal distribution table (Z-table), with Z for 95% confidence level marked

Z	+0,04	+0,05	+0,06	+0,07	+0,08	+0,09
1,7	0,45907	0,45994	0,46080	0,46164	0,46246	0,46327
1,8	0,46712	0,46784	0,46856	0,46926	0,46995	0,47062
1,9	0,47381	0,47441	0,47500	0,47558	0,47615	0,47670
2,0	0,47932	0,47982	0,48030	0,48077	0,48124	0,48169
2,1	0,48382	0,48422	0,48461	0,48500	0,48537	0,48574
2,2	0,48745	0,48778	0,48809	0,48840	0,48870	0,48899

Using all the aforementioned attributes of the network, the sample size ss can be found [Stackexchange, 2017]. First the sample size is found using the normal distribution table together with the chance of the outcome and the confidence interval. The next step is to find the sample size corrected for the entire population ss_{new}.

$$ss = \frac{Z^2 \times p \times (1-p)}{c^2} \tag{A.1}$$

$$ss_{new} = \frac{ss}{1 + \frac{ss-1}{pop}} \tag{A.2}$$

To show that the sample size can be determined using the above two formulas, table A.3 show the results based on a single layer and multi-layer networks with random index vectors.

Table A.3: Results of several network for the determined sample size

Confidence level = 95% ,Z = 1,96 Confidence interval = 2,5% ,c = 0,025 p = 0,5					
Vulnerability Connectivity					
Type of network	Number of times tested	Average vulnerability (4 hits)	confidence interval	% within interval	Within confidence level?
1-Layer	10.000	66,2%	$63{,}7\% \leq x \leq 68{,}7\%$	95,2	Yes
2-Layer	10.000	39,9%	$37{,}4\% \leq x \leq 42{,}4\%$	95,1	Yes
Frigate	500	10,5%	$8\% \leq x \leq 13\%$	99,6	Yes

With the results of the above table, it may indeed be assumed that the sample size can be determined using a normal distribution method. This sample size determination method can be used to find the sample size for different states of the system. Earlier in this chapter, the different states of the system have been discussed. Basically, the different states are determined during one analysis of the network, but not all states have an equal chance to happen. For instance, the state capable of air defence is more prone to failure than the state capable of manoeuvring, this is because the air defence require more systems to function. Assume an $p = 0,5$ for the failure of the air defence, but $p = 0.1$ for the failure of the ability to manoeuvre. This gives the following two sample sizes (with the same confidence level and interval as table A.2):

- $$p = 0.5 \rightarrow ss = 1537$$
- $$p = 0.1 \rightarrow ss = 553$$

The above sample sizes show that a smaller chance of failure gives a smaller sample size. Thus, to determine the sample size for a system in which several states are investigated, the following formula should be used:

$$ss = max(ss_1 ... ss_n), qquad for\ state(1...n) \tag{A.3}$$

B

Appendix B

This appendix contains the hand calculations used for the verification of the method as shown in section 5.1. The calculations made are based whether or not a the requirements have been met. These requirements were:

1. *Part-I Connectivity,*
   Both servers are connected to both the $440V$ distribution network as well as the CW distribution network.

2. *Part-II Capacity,*
   Both servers are supplied with enough electricity and chilled water.

The state of the network was given as a binary. Either the requirements are met and the state is zero, or they are not met and the state is one. This is also shown in the tables given below. Furthermore, the tables show the nodes/edges that has been hit, in case of the 1 hit scenario, seen in table B.1 to B.4 corresponding to the networks of figure 5.2, this is a list followed by the state of the network after the hit. The average vulnerability of the network was then found as:

$$Vulnerability = \frac{\sum states}{n} \tag{B.1}$$

For the two hit cases, the number of states to calculate is larger, as for every network the number of possibilities (n) rises from n to $n \times (n-1)$. This can be seen in table B.5 to B.12. These tables show the state of every combination of two hits, with the first hit vertically and the second hit horizontally. The equation to determine the vulnerability of the networks for these two hit cases stays the same.

Table B.1: Calculations for network 1 with 1 hit

Node	Connectivity	Capacity
1	0	1
2	0	1
3	1	1
4	1	1
5	0	1
6	0	1
7	1	1
8	1	1
9	1	1
10	1	1
Edge		
1	0	0
2	0	0
3	0	1
4	0	0
5	0	1
6	1	1
7	0	0
8	0	1
9	1	1
10	0	0
11	0	0
12	0	1
13	0	0
14	1	1
15	0	0
16	1	1

Table B.2: Calculations for network 2 with 1 hit

Node	Connectivity	Capacity
1	0	1
2	0	1
3	0	1
4	1	1
5	0	1
6	0	1
7	0	0
8	1	1
9	1	1
10	1	1
Edge		
1	0	0
2	0	0
3	0	1
4	0	0
5	0	0
6	0	1
7	0	0
8	1	1
9	1	1
10	0	1
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	1	1
18	0	0

Table B.3: Calculations for network 3 with 1 hit

Node	Connectivity	Capacity
1	0	1
2	0	1
3	0	1
4	1	1
5	0	1
6	0	1
7	1	1
8	1	1
9	1	1
10	1	1
Edge		
1	0	1
2	0	1
3	0	0
4	0	0
5	0	0
6	0	0
7	0	1
8	0	0
9	0	0
10	1	1
11	0	1
12	0	0
13	0	0
14	0	0
15	1	1
16	0	0
17	1	1

Table B.4: Calculations for network 4 with 1 hit

Node	Connectivity	Capacity
1	0	1
2	1	1
3	0	1
4	1	1
5	0	1
6	0	1
7	1	1
8	0	0
9	1	1
10	1	1
Edge		
1	0	1
2	0	0
3	1	1
4	0	1
5	0	0
6	0	1
7	1	1
8	0	0
9	0	1
10	0	0
11	0	0
12	0	0
13	1	1
14	0	0
15	0	0
16	0	0

Table B.5: Calculations for network 1 with 2 hits connectivity

Connectivity																											
	Nodes											Edges															
Node	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		1	1	1	0	0	1	1	1	1		0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1
2	1		1	1	0	0	1	1	1	1		0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1
3	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	1	1		1	1	1	1	1		0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1
6	0	0	1	1	1		1	1	1	1		0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	1
7	1	1	1	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																											
1	0	0	1	1	0	0	1	1	1	1			0	0	1	0	1	1	0	1	0	0	0	0	1	0	1
2	0	0	1	1	0	0	1	1	1	1		0		0	0	0	1	0	0	1	0	0	0	0	1	0	1
3	1	0	1	1	0	0	1	1	1	1		0	0		0	0	1	0	0	1	0	0	0	0	1	0	1
4	1	0	1	1	0	0	1	1	1	1		1	0	0		0	1	0	0	1	0	0	0	0	1	0	1
5	0	0	1	1	0	1	1	1	1	1		0	0	0	0		1	0	1	1	0	0	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1		1	1	1	1	1	1	1	1	1	1
7	1	0	1	1	0	0	1	1	1	1		1	0	0	0	0	1		0	1	0	0	0	0	1	0	1
8	0	0	1	1	1	0	1	1	1	1		0	0	0	0	1	1	0		1	0	0	0	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1
10	0	0	1	1	0	0	1	1	1	1		0	0	0	0	0	1	0	0	1		0	0	1	1	1	1
11	0	0	1	1	0	0	1	1	1	1		0	0	0	0	0	1	0	0	1	0		0	0	1	0	1
12	0	0	1	1	1	0	1	1	1	1		0	0	0	0	1	1	0	0	1	0	0		0	1	0	1
13	0	0	1	1	1	0	1	1	1	1		0	0	0	0	1	1	0	0	1	1	0	0		1	0	1
14	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1		1	1
15	0	0	1	1	1	0	1	1	1	1		0	0	0	0	1	1	0	0	1	1	0	0	0	1		1
16	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table B.6: Calculations for network 1 with 2 hits capacity

Capacity																											
	Nodes											Edges															
Node	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1		1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1		1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																											
1	1	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	0	0	1	0	1	0	1
2	1	1	1	1	1	1	1	1	1	1		1		1	0	1	1	0	1	1	0	0	1	0	1	0	1
3	1	1	1	1	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1		1	0	1		1	1	0	1	1	0	0	1	0	1	0	1
5	1	1	1	1	1	1	1	1	1	1		1	1	1	1		1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1		1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1		1	0	1	1	1	0		1	1	0	0	1	0	1	0	1
8	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1		0	0	1	0	1	1	0	1	1		1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1		0	0	1	0	1	1	0	1	1	1		1	0	1	0	1
12	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1		1	1	1	1
13	1	1	1	1	1	1	1	1	1	1		0	0	1	0	1	1	0	1	1	1	0	1		1	0	1
14	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1		1	1
15	1	1	1	1	1	1	1	1	1	1		0	0	1	0	1	1	0	1	1	1	0	1	0	1		1
16	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table B.7: Calculations for network 2 with 2 hits connectivity

Connectivity

	Nodes										Edges																	
Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1		1	0	1	0	0	0	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
2	1		0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
3	0	0		1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0
4	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	1	1		1	0	1	1	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0
6	0	0	0	1	1		0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0
7	0	0	0	1	0	0		1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
8	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																												
1	0	0	0	1	0	0	0	1	1	1		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
2	0	0	0	1	0	0	0	1	1	1	0		0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
3	1	0	0	1	0	0	0	1	1	1	0	0		0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
4	0	0	0	1	0	0	0	1	1	1	0	0	0		0	0	0	0	1	1	0	0	0	0	0	0	1	0
5	0	0	0	1	0	0	0	1	1	1	0	0	0	0		0	0	0	1	1	0	0	0	0	0	0	1	0
6	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0		0	0	1	1	1	0	0	0	0	0	1	0
7	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0		0	1	1	0	0	0	0	0	0	1	0
8	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0		1	1	0	0	0	0	0	0	1	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
11	0	0	1	1	0	1	0	1	1	1	0	0	0	0	0	1	0	0	1	1		0	0	0	0	0	1	0
12	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0		0	0	0	0	1	0
13	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0		0	0	0	1	0
14	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0		0	0	1	0
15	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	1	1
16	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		1	0
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1
18	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	

Table B.8: Calculations for network 2 with 2 hits capacity

Capacity

	Nodes										Edges																	
Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1		1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	1	0	0	0	1	1
8	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																												
1	1	1	1	1	1	1	0	1	1	1		1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0
2	1	1	1	1	1	1	0	1	1	1	1		1	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0
3	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	1	1	1	0	1		0	1	0	0	1	1	1	0	0	0	0	0	1	0
5	1	1	1	1	1	1	0	1	1	1	0	0	1	0		1	0	1	1	1	1	0	0	0	0	0	1	0
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1		0	1	1	1	0	0	0	0	0	1	0
8	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0		1	1	1	0	0	0	0	0	1	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1
12	1	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1	1		1	0	0	0	1	0
13	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	1	1	1		0	0	0	1	0
14	1	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0		0	0	1	0
15	1	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	0		0	1	1
16	1	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	0	0		1	0
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1
18	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	

Table B.9: Calculations for network 3 with 2 hits connectivity

Connectivity																												
	Nodes											Edges																
Node	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		1	1	1	0	0	1	1	1	1		0	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1
2	1		0	1	0	0	1	1	1	1		1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
3	1	0		1	0	0	1	1	1	1		1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1
4	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	1		1	1	1	1	1		0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
6	0	0	0	1	1		1	1	1	1		0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1
7	1	1	1	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																												
1	0	1	1	1	0	0	1	1	1	1			1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1
2	1	0	0	1	0	0	1	1	1	1		1		0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
3	1	0	0	1	0	0	1	1	1	1		1	0		0	0	0	0	0	0	1	0	0	0	0	1	0	1
4	0	0	0	1	0	0	1	1	1	1		0	0	0		0	0	0	0	0	1	0	0	0	0	1	0	1
5	0	0	0	1	0	0	1	1	1	1		0	0	0	0		0	0	0	1	1	0	0	0	0	1	0	1
6	1	0	0	1	0	0	1	1	1	1		1	0	0	0	0		0	0	0	1	0	0	0	0	1	0	1
7	0	0	0	1	0	1	1	1	1	1		0	0	0	0	0	0		0	0	1	0	0	0	0	1	0	1
8	0	0	0	1	0	0	1	1	1	1		0	0	0	0	0	0	0		0	1	0	0	0	0	1	0	1
9	0	0	1	1	0	0	1	1	1	1		0	0	0	0	1	0	0	0		1	0	0	0	0	1	0	1
10	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1
11	0	0	0	1	0	1	1	1	1	1		0	0	0	0	0	0	0	0	0	1		0	0	0	1	0	1
12	0	0	0	1	0	0	1	1	1	1		0	0	0	0	0	0	0	0	0	1	0		0	0	1	0	1
13	0	0	0	1	0	0	1	1	1	1		0	0	0	0	0	0	0	0	0	1	0	0		1	1	1	1
14	0	0	0	1	0	1	1	1	1	1		0	0	0	0	0	0	0	0	0	1	0	0	1		1	0	1
15	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1
16	0	0	0	1	0	1	1	1	1	1		0	0	0	0	0	0	0	0	0	1	0	0	1	0	1		1
17	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table B.10: Calculations for network 3 with 2 hits capacity

Capacity																												
	Nodes											Edges																
Node	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1		1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1		1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Edge																												
1	1	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1		1	1		0	0	0	1	0	0	1	1	0	0	0	1	0	1
4	1	1	1	1	1	1	1	1	1	1		1	1	0		0	0	1	1	0	1	1	0	0	0	1	0	1
5	1	1	1	1	1	1	1	1	1	1		1	1	0	0		0	1	0	1	1	1	0	0	0	1	0	1
6	1	1	1	1	1	1	1	1	1	1		1	1	0	0	0		1	0	0	1	1	0	0	0	1	0	1
7	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1		1	1	0	1	0	0	1		0	1	1	0	0	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1		1	1	0	0	1	0	1	0		1	1	0	0	0	1	0	1
10	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1		1	1	0	0	0	0	1	0	0	1	1		1	0	1	0	1
13	1	1	1	1	1	1	1	1	1	1		1	1	0	0	0	0	1	0	0	1	1	1		1	1	1	1
14	1	1	1	1	1	1	1	1	1	1		1	1	0	0	0	0	1	0	0	1	1	0	1		1	0	1
15	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1
16	1	1	1	1	1	1	1	1	1	1		1	1	0	0	0	0	1	0	0	1	1	0	1	0	1		1
17	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table B.11: Calculations for network 1 with 2 hits connectivity

Connectivity																											
	Nodes										Edges																
Node	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1		1	1	1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	
2	1		1	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	
3	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	0	0	1	1		1	1	1	1	1	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1	
6	0	0	1	1	1		1	1	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	1	
7	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
9	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Edge																											
1	0	0	1	1	0	0	1	1	1	1		0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	
2	0	0	1	1	0	0	1	1	1	1	0		0	0	0	1	0	0	1	0	0	0	0	1	0	1	
3	1	0	1	1	0	0	1	1	1	1	0	0		0	0	1	0	0	1	0	0	0	0	1	0	1	
4	1	0	1	1	0	0	1	1	1	1	1	0	0		0	1	0	0	1	0	0	0	0	1	0	1	
5	0	0	1	1	0	1	1	1	1	1	0	0	0	0		1	0	1	1	0	0	1	1	1	1	1	
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	
7	1	0	1	1	0	0	1	1	1	1	1	0	0	0	0	1		0	1	0	0	0	0	1	0	1	
8	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	0		1	0	0	0	0	1	0	1	
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	
10	0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1		0	0	1	1	1	1	
11	0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0		0	0	1	0	1	
12	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0	1	0	0		0	1	0	1	
13	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1	0	0		1	0	1	
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	
15	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1		1	
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

Table B.12: Calculations for network 4 with 2 hits capacity

Capacity																											
	Nodes										Edges																
Node	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
6	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
7	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1		1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	0	
9	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Edge																											
1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	0	1	1	1		1	1	0	1	1	0	1	0	0	0	1	0	0	0	
3	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1	1	1	1	0	1	1	1	0	1	1		1	1	1	1	0	0	0	1	0	0	0	
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1		1	0	0	0	1	0	0	0	
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1		1	0	1	0	0	0	
11	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1		0	1	0	0	0	
12	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0		1	0	0	0	
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	
14	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1		0	1	
15	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0		0	
16	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0		

Bibliography

R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47 – 97, January 2002.

D. Andrews. Preliminary warship design. *Trans. RINA*, 136(A):37–55, 1994.

S. Boccaletti, J. Buldú, R. Criado, J. Flores, V. latora, J. Pello, and M. Romance. Multiscale vulnerability of complex networks. *Chaos*, 17, 2007.

D.K. Brown. The future british surface fleet: Options for medium-sized navies. *Conway maritime press*, 1991.

R. Cohen and S. Havlin. *Complex Networks: Structure, Robustness and Function*. Cambridge University Press, 2010.

P. de Vos. On the application of network theory in naval engineering, generating network topologies. *INEC*, 2014.

P. de Vos. Automatic topology generation as decision support tool in early design stages of on board energy distributions systems. To be published PhD thesis, 2017.

P. de Vos, D. Stapersma, and B.J. van Oers. Automatic topology generation as decision support tool in early design stages of on board energy distributions systems. To be published paper, 2017.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. on Evolutionary Computation*, 6(2):182 – 197, 2002.

Dictionary.com. Redundancy (engineering). `http://www.dictionary.com/browse/vulnerability`, 2017. Visited on 23 November 2016.

E.A.E. Duchateau. Interactive evolutionary concept exploration in preliminary ship design. Phd thesis, TU Delft, Delft, The Netherlands, 2016.

A. Dwivedi, X. Yu, and P. Sokolowski. Analyzing power network vulnerability with maximum flow based centrality approach. *International Conference on Industrial Informatics*, 2010.

S.O. Erikstad. A decision support system for preliminary ship design. Phd thesis, NTNU, Trondheim, Norway, 1999.

I. Eusgeld, W. Kröger, G. Sansavini, M. Schläpher, and E. Zio. The role of network theory and object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures. *Reliability Engineering and System Safety*, 94, 2009.

C. Gollub and R. de Vivie-Riedle. Multi-objective genetic algorithm optimization of 2d- and 3d- pareto fronts for vibrational quantum processes. *New Journal of Physics*, 11, 2009.

Z. Guohua, W. Ce, Z. Jianhua, Y. Jingyan, Z. Yin, and D. Manyin. Vulnerability assessment of bulk power grid based on complex network theory. *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, 2008.

F.S. Hillier and G.J. Lieberman. *Introduction to operations research*. McGraw-Hill, New York, the United States of America, 2010.

P. Holme, B.J. Kim, C.N. Yoon, and S.K. Han. Attack vulnerability of complex networks. *Physical review*, 65, May 2002.

A. Kunjur and S. Krishnamurty. Genetic algorithms in mechanism synthesis. *Journal of Applied Mechanisms and Robotics*, 4(2):18 – 24, 1997.

G. Mao and N. Zhang. Analysis of average shortest-path length of scale-free network. *Journal of Applied Mathematics*, 2013, 2013.

I. Mishkovski, M. Biey, and L. Kocarev. Vulnerability of complex networks. *Communications in Nonlinear Science and Numerical Simulation*, 2010.

M.E.J. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167 –256, 2003.

J. Parent. Vulnerability reduction of naval ships. Presentation, 2015.

D.T. Rigterink. *Methods for analyzing early stage naval distributed systems designs, employing simplex, multi-slice, and multiplex networks*. PhD thesis, University of Michigan, 2014.

Stackexchange. Sample size choice with binary outcome. `https://stats.stackexchange.com/questions/207584/sample-size-choice-with-binary-outcome`, 2017. Visited on 8 august 2017.

G. van Ingen. Een methodiek voor het reduceren van de kwetsbaarheid van marine schepen. Msc thesis, TU Delft, Delft, The Netherlands, 2011.

B.J. van Oers. A packing approach for the early stage design of service vessels. Phd thesis, TU Delft, Delft, The Netherlands, 2011.

B.J. van Oers, G. van Ingen, and D. Stapersma. An integrated approach to design, route and evaluation of resilient ship service systems. *INEC*, 2012.

Wikipedia. Redundancy (engineering). `https://en.wikipedia.org/wiki/Redundancy_(engineering)`, 2017. Visited on 17 April 2017.

S. Xu, H. Zhou, C. Li, and X. Yang. Vulnerability assessment of power grid based on complex network theory. *Asia-Pacific Power and Energy Engineering Conference*, 2009.

H. Yang, Z. Ye, D. Mei, and H. Xiao. Vulnerability assessment of shipboard electric power information network. *WARTIA 2016*, 2016.