The effects of network topology on energy networks

Master’s thesis
January 1, 2017
Student: Julien van der Land
Primary supervisor: Prof. dr. ir. Marco Aiello
Secondary supervisor: MSc. Laura Fiorini
Abstract

Energy grids are shifting from centralized and predictable fossil based power generation towards more sustainable sources such as solar and wind. Inherent to generating electricity from renewable sources is the volatility of such generation, this volatility introduces new challenges for the transmission grid operators to balance the grid and guarantee secure operation. Due to patterns in weather the power output of renewable generators cannot be perfectly predicted, nor can the demand of consumers. These uncertainties can introduce imbalance into the system. In order to resolve this imbalance as a last resort the transmission system operator can shed load from consumers or curtail power generation from renewable sources.

A way to alleviate the effects of the volatility is to introduce energy storage systems into the grid. These systems can be charged when there is an excess in power generation, by storing the energy surplus, and discharged during periods of high demand, reducing the risk of Expected Energy Not Supplied (EENS). This leads to a better utilization of renewable power sources over time and a more secure power supply for consumers. In order to determine the feasibility of a transmission grid with energy storage systems we have developed a program that simulates several realistic scenarios using Monte Carlo techniques.

The aim of this work is to determine the optimal size and location of storage devices to reduce the EENS and reduce congestion on the most utilized lines. Moreover we compare the results across 3 different topologies, i.e. a modified IEEE-96 bus, a small world topology and a preferential attachment topology. For each network we place the storage according to three different siting policies. We determine the optimal size by comparing the investment costs of storage and the economical values of their benefits.

The investment costs of storage are calculated according to three different pricing profiles and two different energy to power ratios. In case of the cheapest prices, we find that the storage is economically viable with a size up to 30% and 20% of the installed renewable generation capacity, depending on the energy to power ratio. In contrast, in the most expensive scenario a relatively small storage size is still feasible when compared to the savings gained from EENS reduction.

Regarding the siting, placing storage near consumers is the most effective in reducing EENS while different policies do not significantly change the power flow. However we do find significant cases to discuss on the modified IEEE-96 bus and the small world topology.
# Contents

1 **Introduction** .................................................. 1
   1.1 Research questions ........................................... 1
   1.2 Methodology ................................................... 2
   1.3 Thesis organization .......................................... 2

2 **Related Work** ................................................ 4
   2.2 Hybrid solar/wind power system probabilistic modeling for long-term performance assessment - Tina et al (2005) .................. 4
   2.4 The integration of storage in HV grids: optimal use of renewable sources - Fiorini (2014) .................................................. 7
      2.4.1 Renewable power curtailment .............................. 7
      2.4.2 Energy Storage Systems .................................... 8
      2.4.3 Linearized power systems model ......................... 8
      2.4.4 Modified IEEE reliability test system 1996 ............ 8
      2.4.5 Results Fiorini et al ...................................... 10
      2.5.1 The value of supply security The costs of power interruptions: Economic input for damage reduction and investment in networks - Nookj et al (2007) .................. 11
   2.6 The Power Grid as a complex network: A survey - Pagani and Aiello (2013) .................................................. 12

3 **Background** .................................................... 13
   3.1 Definitions ...................................................... 13
   3.2 The grid explained .............................................. 13
      3.2.1 Edges ....................................................... 14
      3.2.2 Nodes ....................................................... 14
   3.3 Topological representation of a power grid ................... 16
      3.3.1 Definitions specific to the power grid ............... 17
   3.4 Different types of graphs ..................................... 18
      3.4.1 Small world graph ....................................... 18
      3.4.2 Preferential attachment graph ....................... 19
List of Figures

2.1 Modified version of the IEEE reliability test system 1996 used by Fiorini [12] ........................................ 9
3.1 Small world topology with distributed storage .......................................................... 19
3.2 Preferential attachment topology with distributed storage ......................................... 20
4.1 Flow of the program ........................................................................................................ 28
4.2 Class diagram of the program ...................................................................................... 29
4.3 Class diagram of the model package ............................................................................. 31
4.4 Class diagram of the graph package ............................................................................ 33
4.5 Class diagram of the fileHandler package ................................................................. 34
4.6 Class diagram of the simulation package .................................................................... 35
4.7 Class overview of the Grid Visualization tool ............................................................. 36
4.8 GUI class ....................................................................................................................... 37
4.9 GraphGenerator class .................................................................................................. 38
4.10 The GraphLogic class ................................................................................................. 38
4.11 The GraphMetrics class ............................................................................................... 39
4.12 Grid Visualization Tool Inspection ........................................................................... 40
4.13 Grid Visualization Tool following the initial generation of a small world graph ............ 42
6.1 IEEE 96 modified grid with distributed storage ........................................................... 50
6.2 Sizing of storage with price profile 1 .......................................................................... 53
6.3 Storage sizing with price profile 2 .............................................................................. 54
6.4 Sizing of storage with price profile 3 .......................................................................... 55
6.5 Line exploitation of the modified IEEE topology without storage: Maximum usage count: 4344 .................................................. 59
6.6 Line exploitation of the small world topology without storage: Maximum usage count: 7128 .................................................. 60
6.7 Line exploitation of the preferential attachment topology without storage: Maximum usage count: 4176 .................................................. 61
6.8 Line utilization of the modified IEEE-96 topology with storage near consumers: Maximum usage count: 4440 .......................................................... 64
6.9 Line utilization of the small world topology with storage near consumers: Maximum usage count: 4248 .......................................................... 65
6.10 Line utilization of the preferential attachment topology with storage near consumers: Maximum usage count: 5520 .......................................................... 66
6.11 Line utilization of the modified IEEE-96 topology with storage near lowest PTDF node: Maximum usage count: 6600 .......................................................... 69


6.15 Line utilization of the small world topology with storage near the node with highest centrality betweenness. Maximum usage count: 3744.

6.16 Line utilization of the preferential topology with storage near least contributing node. Maximum usage count: 7272.
List of Tables

2.2 Average VoLL for firms, government, and households over 9 periods for different times and days ........... 12
4.1 Key layout of the Grid visualization tool ................................................................. 41
6.1 Sizing profiles of Storage and their Savings .......................................................... 50
6.2 Return from using storage in percentage of investment using price profile 1 ....................................... 51
6.3 Return from using storage in percentage of investment using price profile 2 ....................................... 51
6.4 Return from using storage in percentage of investment using price profile 3 ....................................... 51
6.5 Annual cost of EENS using constrained grids without storage ......................................................... 56
6.6 10 most utilized lines of the modified IEEE 96 topology without storage .......................................... 57
6.7 10 most utilized lines of the small world topology without storage ....................................................... 57
6.8 10 most utilized lines of the preferential attachment topology without storage ...................................... 58
6.9 Annual cost of EENS using constrained grids with distributed storage near consumers. The baseline we refer to can be found in table 6.5 ................................................................. 62
6.10 10 most utilized lines of the modified IEEE 96 topology with distributed storage near consumers ................ 62
6.11 10 most utilized lines of the small world topology with distributed storage near consumers .................. 63
6.12 10 most utilized lines of the preferential attachment topology with distributed storage near consumers ... 63
6.13 Annual cost of EENS using constrained grids with distributed storage lowest Power Transfer Distribution Factor (PTDF) node. The baseline we refer to can be found in table 6.5 ................................................................. 67
6.14 10 most utilized lines of the modified IEEE 96 topology with storage near lowest PTDF node ................ 67
6.15 10 most utilized lines of the small world topology with storage near lowest PTDF node ....................... 68
6.16 10 most utilized lines of the preferential attachment topology with storage near lowest PTDF node .... 68
6.17 Annual cost of EENS using constrained grids with centralized storage at the node with the highest centrality betweenness. The baseline we refer to can be found in Table 6.5.

6.18 10 most used lines of the modified IEEE 96 topology with a storage node placed at the node with highest centrality betweenness.

6.19 10 most used lines of the small world topology with a storage node placed at the node with highest centrality betweenness.

6.20 10 most used lines of the preferential attachment topology with a storage node placed at the node with highest centrality betweenness.

6.21 Metrics of the modified IEEE-96 bus shown in Fig. 6.1.

6.22 Metrics of the small world topology shown in Fig. 3.1.

6.23 Metrics of the preferential attachment topology shown in Fig. 3.2.
Chapter 1

Introduction

Electricity networks can be represented as complex graphs, analysis of these graphs can help us learn more about these networks. Using knowledge gained from this analysis we can plan for more effective networks. Additionally this analysis allows us to more effectively plan maintenance, or extensions to the electricity network. With recent advancement in energy storage technology and the advent of renewable power generation from sources such as wind and solar, it has yet to be determined how these different components come together. That is to say; what would the future electricity grid look like given the volatile nature of renewable energy sources and the capability of storing energy for later use? What would be an effective topology for an electricity network using these elements? And how do these elements impact the operation of such an electricity network? This work is intended to answers some of these questions.

1.1 Research questions

Recently there has been a drive from governments, citizens, and Non-Government Organization (NGO) to a more sustainable energy policy in order to limit the impact of climate change. Part of this drive has been to push electricity generation away from fossil fuels and towards renewable energy sources such as wind and solar power. An example of this drive is the 2020 climate & energy package [1] of the European Union which states that 20% of the energy generated in the European union should be from renewable sources by 2020. Another example is the Paris Agreement which was signed by 190 countries. This drive towards using renewable energy sources comes with challenges for energy grids due the volatile nature of these sources. Variations in wind speed and cloud cover change the power output of respectively wind turbines and solar panels, we therefore call these sources volatile. The volatility of these sources introduces uncertainty in the amount of power being injected into the grid. This can be addressed by using Energy Storage System (ESS) which can charge when the output of renewable sources is high, and discharge when the output is low. The introduction of ESS comes with its own challenges for an energy grid. Therefore in this work we are concerned with the following questions;

• What is the optimal size and location of the storage?
• How does the topology of the network influence operation of the storage?

1.2 Methodology

In order to answer the questions posed in section 1.1 we developed a program that simulates the DC power flow on a given topology. The design of this program and an accompanying tool to visualize and generate topologies are presented in this thesis. For this work we have used a modified IEEE-96 bus along with two other generated topologies, whose structure and generation process are explained in detail. In addition we also present an extensive analysis of the results following the execution of the program. We determine the performance of the topologies in terms of Expected Energy Not Supplied (EENS) and how the power flow on transmission lines are influenced following the introduction of storage.

1.3 Thesis organization

This thesis is organized in 9 major chapters. What follows is a short explanation of the contents in each chapter.

In Chapter 2 we review the papers that were most influential for this work, detailing which aspects were more relevant and why. In this chapter we also present the modified IEEE-96 bus used in this work.

Chapter 3 introduces definitions and concepts that are important for this work. Examples of this include the definitions of edges and nodes. We also detail several metrics such as EENS and we give the definitions related to the topological representation of a power grid. Finally we also present the small world and preferential attachment topologies that we have used in this work.

Chapter 4 presents the overall concepts that were used in the simulation and grid visualization programs. We detail the conceptual steps through which the simulations run and we explain the policies that govern different aspects of the simulation such as; the operation of consumers, generators, etc. Moreover for the simulation program and grid visualization tool we also detail the architectural design that was used from a software engineering perspective. That is to say we explain the different classes of the simulation program and grid visualization tool and how these classes are related to each other.

Chapter 5 introduces the mathematical formulation of the simulation program. We present the mathematical models required for the operation of wind and solar farms, conventional generators, consumers, storages and transmission lines. Additionally we also report the model that was used by the linear program to compute the power flow.

In Chapter 6 we report the results achieved in terms of determining the size and siting of the storage. We present 10 different sizing profiles and three pricing profiles. An economical evaluation is done using these profiles in order to determine the optimal storage size. Moreover we report the performance of
different topologies before and after the storage introduction, following three storage siting policies. For each of the three topologies we present the results of the siting policies. In addition we also present several metrics that we have used to differentiate between the topologies.

In Chapter 7 we draw our conclusions regarding the research questions from section 1.1 by analyzing the results from Chapter 6.

In Chapter 8 we discuss the results and conclusion from respectively Chapter 6 and Chapter 7. In this discussion we offer explanations why certain results were achieved or why certain conclusions were drawn. The topics that are covered in this chapter are; Sizing of storage, the influence of the siting policies on storage performance and, the influence of topology on EENS.

Chapter 9 is the final chapter of this work, in it we present possible future works.
Chapter 2
Related Work

In this chapter we present an overview of literature that are relevant to this work. For each of the papers that we have selected we present an overview of the relevant sections and we detail which concepts we have have and have not used.


Arianos et al [7] investigated the topological structure and resilience of power grids. The authors introduced a new distance measure that not only takes into account the topological path between two nodes but also takes into account the PTDF and the impedance of a line, i.e. the physical properties of the grid. Using this new distance measure they computed the novel net-ability metric. The net-ability metric estimates the performance and resilience of a grid when subject to removal of a line. Arianos et al uses three different methods to evaluate the impact of line outages: 1) A method based on efficiency, 2) Their new net-ability metric, and 3) Computation of line overloads by DC power flow. Due to the confidentiality that governs power grid vulnerability explicit validation of these methods is not available. Arianos et al have reported that based on direct observations from the Italian power grid operator a good match has been found using their new net-ability metric in terms of grid vulnerability. For our own work we have used the definitions described by Arianos et al of the admittance matrix, the transmission matrix, and the PTDF. We use the PTDF for placing storage nodes at different positions depending on the policy.

2.2 Hybrid solar/wind power system probabilistic modeling for long-term performance assessment - Tina et al (2005)

Tina et al presents a probabilistic method based on convolution techniques to assess the long term performance of hybrid solar-wind power systems for standalone and grid linked applications. The authors performed a reliability
analysis by use of the energy index of reliability which is directly related to EENS. The model used for their simulation is based on a wind energy conversion system that is connected in parallel to a Photovoltaic System (PVS). The grid is assumed to be bi-directional and excess of energy is conditionally supplied to the grid. Deficits of energy are drawn by the grid in the low generating phase to supply local demand.

In order to simulate the Wind Energy Conversion System (WECS) the authors used a random variable that is drawn from a Weibull distribution. This random variable resembles the wind speed. The output of the WECS is then computed using cut-in-speeds (wind speed at which the turbine starts turning), and cut-out-speed (winds speed that the turbine was not designed for). The authors assumed a linear relationship between the wind-speed and the power generated by the WECS. They also note that there is a linear relationship between the amount of solar irradiance that reaches a PVS and the power output of said PVS. Many factors affect the amount irradiance such as the geographical location, time, and the climate conditions. According to Tina et al many studies have proved that the cloudiness is the main factor affecting the solar irradiance measured inside the atmosphere. Therefore when calculating the output of a PVS they take into account the following: The surface area of the PVS, the irradiance of a surface given an inclination, and the efficiency of the PVS. The PVS considered by the authors are assumed to be equipped with Maximum Power Point Tracker (MPPT). The authors validate their approach by comparing the energy index of reliability produced by their analytical model with that same metric produced from a previously developed Monte Carlo Simulation using MatLab-Simulink. They find that given enough simulations the Monte Carlo Simulation converges to that of the probabilistic model.

It should be noted that Tina et al developed an analytical method in contrast to the simulation based method that we are using. The difference being that in the analytical case the system is represented as a mathematical model from which its reliability incidences are computed in the form of direct solutions. In our case the underlying processes that produce these solutions are computed by modeling them. We use probabilistic methods, which treat certain variable such as wind speed, cloud cover, load, and so on as probability distributions. Our approach to calculating the output of PVS is based on the work of Tina et al but with some differences; the output of a PVS is given as $A_c \cdot \eta \cdot I_\beta$, where $A_c$ is the array surface in $m^2$, $\eta$ the efficiency of the PVS, $I_\beta$ the irradiance to the horizontal plane $\beta$. The definition of the irradiance $I_\beta$ from [21] has been replaced in favor of the definition of irradiance as specified by [19]. This was done because the model for the irradiance presented in [19] was more suited to our Monte Carlo approach. [19] uses sinusoidal waves, and parameters representing the variance of solar irradiance and the frequency of weather changes thus computing the irradiance $I_\beta$ taking into account weather (Eq. 4 from [19]). For our purposes we have instead represented the cloudiness as a Monte Carlo Draw (MCD) from a Gamma distribution as specified by [9].

Pandzic et al proposed a framework for the optimal sizing and siting of ESS distributed across a transmission grid. This framework is based on a 3 stage process:

1. At every bus a storage of unlimited size and power rating is assumed. They run a simulation for a 24 hour period with an objective function that minimize the cost of power generation and the daily investment cost of the storage. The most optimal siting for the storage is then identified by identifying the most used storages.

2. Using the locations identified at stage 1 they once again run the simulation, assuming storage of unlimited size and power rating. The maximum energy stored and power injected than decide the energy and power capacities for each storage unit on a per day basis. These values are then averaged and passed to stage 3.

3. At this stage the simulation is run again, but this time with fixed storage size, power rating and location. The results of this stage indicate the benefits that can be achieved by deploying different amounts of storage.

The authors performed a case study using this framework and an updated IEEE RTS-96 bus. The wind power production was simulated over a whole year with hourly resolution based on data from the NREL Western Wind dataset. They normalized the dataset by subtracting the average for the corresponding month and diving by the standard deviation for the corresponding month. The data was then detrended and transformed into stationary Gaussian distributed series. The following time series models are then fitted to this normalized data: AR(2), AR(3), ARMA(2,1), ARMA(3,1), and ARMA(3,2). Each model is updated every 6 hours, providing a new 6 hour prediction. Spatial correlation is added by using a covariance matrix that generates spatially correlated noise. For each model 100 estimates are generated based on this random noise, resulting in 500 estimates every 6 hours. An inverse transformation is then applied and the trend is added, producing the final wind speed data. The wind speed is then converted into wind power using a power curve derived from the original dataset.

Pandzic et al analyzed 3 different cost profiles for the storage:

1. 20$/kWh and 500$/kW per unit of storage
2. 50$/kWh and 1000$/kW per unit of storage
3. 100$/kWh and 1500$/kW per unit of storage

The expected lifetime of the storage was set to 20 years, the interest rate at 5%. The round-trip efficiency of the storage was set to 0.81. Based on cost profile 1 they reduce the cost of power generation by 2.46%, profile 2 show a reduction
of 0.16%, the investment cost of profile 3 was too high, causing the storage not to be used.

The authors also performed a sensitivity analysis as the storage should be insensitive to small deviations in wind output. They compared the results with scenarios where wind output was 5% lower, 1% lower and 5% higher, 1% higher. They found that the storage is insensitive to these small changes. In addition to the sensitivity analysis Pandžić et al also analyzed the impact of congestion. They found that when the grid is not congested their methods can be used to determine storage capacity, when congestion occurs however the location of the storage is better determined by other factors.

From Pandžić et al we have used the models related to calculating the investment cost of the storage, the pricing profiles were also used, along with the lifetime of the storage and the interest rate. While [17] used a round-trip efficiency of 0.81 the round-trip efficiency of our storage was set to 0.75. Our approach for finding the sizing of the storage is somewhat similar to [17] but there are also some key differences. We also used an unconstrained grid when deciding the size of the storage, but in our case the objective function is based on reducing curtailment and load shedding. We examined different storage sizes based on a percentage of the total renewable power generation capacity, instead of calculating them based on storage usage.

2.4 The integration of storage in HV grids: optimal use of renewable sources - Fiorini (2014)

Fiorini et al examined the benefit of introducing ESSs to a power system with a high penetration of renewable power generation. They represent the electricity grid as a weighted graph which is used to simulate the DC power flow using an objective function that minimizes curtailment of renewable energy and production costs of conventional generators. This was achieved by developing a program which computes the power flow by representing the production-supply problem as a linear programming problem of which the variables are subjected to constraints. The following are key concepts from this paper on which this work is based.

2.4.1 Renewable power curtailment

Renewable power curtailment is the act of cutting the injection of renewable power into the grid. Because of the low marginal cost associated with wind and solar energy their curtailment is generally considered undesirable. Regardless there are of course situations in which their curtailment is needed. Congested transmission lines, minimum technical power generation of thermal generators [12] are example of reasons for renewable power curtailment. The objective function of the linear program that Fiorini et al used aims to minimize the curtailment of renewable power and the cost of production of conventional generators.
2.4.2 Energy Storage Systems

There are many different types of ESSs each with different properties. Fiorini et al listed the following different ESSs: Pumped hydro storage, compressed air energy storage, flywheels, superconducting magnetic energy storage, capacitors, and batteries. Each of these different ESS technologies have specific properties in terms of efficiency, storage capacity, rate of (dis)charge, etc. Fiorini et al only considered battery storage in their work without drawing distinctions between different types of battery storage.

2.4.3 Linearized power systems model

We consider the full linearized power system model used by Fiorini et al to be outside the scope of this thesis. Suffice it to say that the model is based on a nonlinear AC model that shows sinusoidal behavior. Since such an nonlinear model is difficult to solve it is simplified to a set of linear equations called DC power flow. We refer to [12] [7] and [13] for a full explanation of the DC power flow. The simulation of the power flow for our work is based upon that of Fiorini et al.

2.4.4 Modified IEEE reliability test system 1996

Fiorini et al performed an evaluation of the program they developed using a modified version of the IEEE reliability test system 1996 [12]. The total production amounts to 10215 MW and the peak load is assumed to be 8550 MW [12]. In addition Fiorini et al added 10 storage nodes, 15 wind farms, and 12 photovoltaic solar farms. Finally [12] reduced the capacity of the lines to 75% of their original values in order to study the networks behavior under critical state. In fig. 2.1 we can see the modified version with additional transmission lines, storage and renewable generators.

For this work we have used the modified IEEE system created by Fiorini et al in order to perform the sizing of storage. In addition to that we have also used this network for testing, simulation and analysis. The basis of the modified IEEE-96 bus is also used to generate topologies with specific properties.
Figure 2.1: Modified version of the IEEE reliability test system 1996 used by Fiorini [12]
2.4.5 Results Fiorini et al

Fiorini et al found that the introduction of ESSs allows the storage to charge during off-peak periods or when there is an excess of renewable energy. The introduction of ESSs allowed the production of conventional generator to be lowered by utilizing the storage. The authors also found that the introduction of ESSs into a network lowered the curtailment of renewable energy. In addition, Fiorini et al found that when ESSs are present in the network the congestion of lines is reduced. The siting of the ESSs did not have a significant impact on the congestion of the lines. Lastly, the authors found that the introduction of ESSs does not change which lines carry the highest portion of the global power flow.


According to Humphries and Gurney the original small-world property of [22] is a categorical one and does not allow one to measure to what extent a graph has the small-world property. Therefore Humphries and Gurney introduced a new measure that is continuous and quantitative. First let us define the characteristic path length [20] [15].

\[ L = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} \]  

(2.1)

Where \( d_{ij} \) is the shortest geodesic between node i and j.

Then let us define the clustering coefficient at an individual node: [22].

\[ C_{ws} = \frac{2E_i}{k_i(k_i-1)} \]  

(2.2)

Let \( E_i \) be the number of edges between the neighbors of i and let \( k_i \) be the degree of nodes i. The clustering coefficient \( C \) is then the mean of \( C_{ws} \).

Let the new measure of the small-world property be:

\[ \sigma = \frac{C/C_{rand}}{L/L_{rand}} = \frac{\gamma}{\lambda} \]  

(2.3)

Where \( C_{rand} \) and \( L_{rand} \) are the clustering and characteristic path length of a random graph with the same amount of nodes and edges as the graph that we are considering. Then following definition 2 from [15] a network is said to be a small-world network if \( \sigma > 1 \). Humphries and Gurney find that the Watts-Strogatz model [22] with a constant rewiring parameter and \( \sigma \) will scale linearly with \( n \). The authors examined 33 networks and in many cases these networks also exhibit such linear scaling. They have also shown that the linear scaling between \( \sigma \) and \( n \) is not inevitable, networks with large edge density do not exhibit such linear scaling. From [15] we have used the definition of eq. (2.3) to determine if a graph is small-world.
2.5.1 The value of supply security: The costs of power interruptions: Economic input for damage reduction and investment in networks - Nooij et al (2007)

Nooij et al examined the impact of what is referred to as Value of Lost Load (VoLL). VoLL is the economical impact of not supplying electricity to consumers. The economical valuation of VoLL is done using the production-function approach. This approach estimates the consequences of power outages through lost production for businesses and governments or lost time for households. The authors directly estimated the lost production of each sector and aggregated these estimations into a macro-economical total. Because the authors assumed that during an outage all activities grind to halt they implemented sector linkage by manipulating input-output tables. By doing so they ensure that an interrupted activity can be not distributed further by nonproduction of another sector. According to Nooij et al businesses and governments suffer three kinds of damages when energy is not supplied: lower production, lost production, and cost caused by restarting production. The authors only considered the cost associated with lost production during energy supply interruptions. The cost incurred by households is based on two different types; loss of leisure time, and the loss of goods. They assumed that during an electricity supply interruption all leisure is lost. The value of leisure is determined by the assumption that 1 hour of work equals 1 hour of leisure. Using this assumption and a breakdown of the activities people do the authors calculated the value of leisure time. The values for losses from leisure, businesses and government are based on data from the Netherlands.

<table>
<thead>
<tr>
<th>Sector</th>
<th>Value of Lost Load in kWh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>3.9</td>
</tr>
<tr>
<td>Energy sector</td>
<td>-0.32</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>1.87</td>
</tr>
<tr>
<td>Construction</td>
<td>33.05</td>
</tr>
<tr>
<td>Transport</td>
<td>12.42</td>
</tr>
<tr>
<td>Services</td>
<td>7.94</td>
</tr>
<tr>
<td>Government</td>
<td>33.5</td>
</tr>
<tr>
<td>Firms and government</td>
<td>5.97</td>
</tr>
<tr>
<td>Households</td>
<td>16.38</td>
</tr>
<tr>
<td>Firms, government and households</td>
<td>8.56</td>
</tr>
</tbody>
</table>

Table 2.1: Value of lost load for the Netherlands in euros for the energy sector [11]

Nooij et al reported that the VoLL differs depending on the exact timing of an outage in terms of the hour, weekdays and weekdays. In order to determine a average for weekdays and weekend days the authors calculated the average VoLL using data from nine different periods. These averages capture variations of sectors that are active during different times. As we can see in table 2.2 there are some significant differences in VoLL depending on the time and day outages occur.
<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>VoLL in euros per kWh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekdays</td>
<td>Day (08:00-18:00)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>evening (18:00-24:00)</td>
<td>8.9</td>
</tr>
<tr>
<td></td>
<td>night (24:00-08:00)</td>
<td>2.7</td>
</tr>
<tr>
<td>Saturdays</td>
<td>Day (08:00-18:00)</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>evening (18:00-24:00)</td>
<td>12.5</td>
</tr>
<tr>
<td></td>
<td>night (24:00-08:00)</td>
<td>3.9</td>
</tr>
<tr>
<td>Sundays</td>
<td>Day (08:00-18:00)</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>evening (18:00-24:00)</td>
<td>12.5</td>
</tr>
<tr>
<td></td>
<td>night (24:00-08:00)</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Table 2.2: Average VoLL for firms, government, and households over 9 periods for different times and days

Since the calculations of the sizing for the storage requires a valuation of EENS. We chose the national average VoLL for firms, government and households from table 2.1. In our simulation we do not take into account weekdays and weekends, instead we simulate one day of a season, which is then summed and multiplied by a coefficient in order to represent the result of an entire year. For this reason the average VoLL for firms, governments and households from table 2.1 has been used to valuate EENS as opposed to the VoLL from table 2.2.

2.6 The Power Grid as a complex network: A survey - Pagani and Aiello (2013)

The performance of an energy grid regarding its resilience to attacks, and failures is in large part dependent on the topology of such a grid. It is therefore important to consider the topological properties of an energy grid when examining its performance during specific events. A large volume of literature exists that does exactly this; they analyze the performance of an energy grid during specific events such as a cascading failure, attacks from terrorism, etc. Pagani and Aiello presented a survey of several papers and summarized the different properties that are looked at in these papers. The authors find that many of the papers examine the small-world property, and the path length property. In addition many of the papers also do some form of resilience analysis. The node degree distributions statistics and betweenness distribution statistics are less common. [16] notes that it is important to represent the physical properties in terms of impedance for the nodes, and edges in graph because it increases the realism of the simulation/analysis. We consider the work done by Pagani and Aiello to be an excellent introduction for network analysis of power grids and many of the concepts and properties described in [16] are relevant for our own work.
Chapter 3

Background

In this chapter we introduce the definition of **EENS** and concepts that are important for this work which were not featured in chapter 2. The definition of **EENS** is described in section 3.1 followed by section 3.2 which explains the concepts that are important for an energy grid. In section 3.4 we introduce two synthetic topologies that have been used in this work and we explain the basis of these topologies. Section 3.5 introduces and explains the economical model that was used for the sizing of storage.

3.1 Definitions

**Definition 3.1.1** [Expected Energy Not Supplied]. Expected Energy Not Supplied measures whether enough electricity is being supplied to sink nodes from source nodes. There are many reasons why Expected Energy Not Supplied occurs, among them are generator failures, congestion of lines, or poor production planning.

\[ \text{EENS} = \sum_{c \in C} \text{load}_c - \sum_{e_{v_c,v_i}} f_e \]  

(3.1)

Where \( f_e \) is the flow on edge \( e \) connecting consumer \( v_c \) to inner node \( v_i \). And \( \text{load}_c \) is the load of consumer \( c \).

**Definition 3.1.2** [Degree distribution]. The degree distribution of graph expresses the fraction of vertices with degree \( k \). Let \( K_{\text{max}} \) be the maximum degree in graph \( G \).

\[ K_{i=1,2,..,K_{\text{max}}} = \sum_{j \in E} \text{degree}_j \]  

(3.2)

Where \( \text{degree}_j \) is the degree of edge \( j \) if that degree is equal to \( i \).

3.2 The grid explained

The energy grid is represented by a graph made up of edges and nodes. Before describing the general procedure of the program it is worth, to first describe the different attributes of edges and nodes that exist on the grid. Our definition of the power grid represented as a graph is very similar as the one presented in [12].
3.2.1 Edges

Edges represent the transmission lines between two nodes. They have the following attributes:

- **Capacity**: The maximum power that can flow over a line.
- **Reactance**: The imaginary part of the impedance. The reactance is included in the calculation of the load flow because it results in a more realistic simulation.

Like [12] we must introduce the notion of real and virtual edges; a real edge is defined by its reactance, which gives weight to the edge and capacity which determines how much power can pass through the edge. A virtual edge is defined by a capacity that is always large enough to carry the load to a consumer, or from a producer. In addition virtual edges have a very low reactance $10^{-4}$ per unit, therefore we can say that real edges are weighted and virtual edges are unweighted.

3.2.2 Nodes

- **Conventional generators**

  The conventional generators have the following attributes:

  - **Mean Time To Failure**
    
    The average duration it takes for a generator to fail
  
  - **Mean Time To Repair**
    
    The duration it takes to repair a generator after failure
  
  - **Maximum Production change**
    
    Limits the production increases of conventional generators due to their spinup time; the maximum increase or decrease is 50% of the maximum production
  
  - **Maximum production**
    
    The maximum production capacity of a generator in MW
  
  - **Minimum production**
    
    The minimum production capacity of a generator in MW
  
  - **Day ahead limit maximum**
    
    Limits the maximum production to 7.5% below the maximum production capacity
  
  - **Day ahead limit minimum**
    
    Limits the maximum production to 7.5% above the minimum production capacity

Conventional generators can represent the following generator subtypes: nuclear, oil, coal, and hydroelectric.
The maximum and minimum production capacity is defined for each generator and so their values can differ for each generator. The day ahead production maximum and day ahead limit minimum are defined globally, i.e., every conventional generator has the same day ahead production maximum and day ahead limit minimum. Like day ahead production maximum and day ahead limit minimum, the Mean Time To Repair (MTTR) and Mean Time To Failure (MTTF) are also applied globally, it should be noted however that hydroelectric generators are exempt from failure, and as such they do not have a MTTR and MTTF.

- **Wind generators**
  Following the approach of [12], the wind generator nodes simulate Vestas V90-2.0MW turbines with the following properties:

<table>
<thead>
<tr>
<th>Rated power</th>
<th>2MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCutIn</td>
<td>3 m/s</td>
</tr>
<tr>
<td>vCutOff</td>
<td>25 m/s</td>
</tr>
<tr>
<td>vRated</td>
<td>12 m/s</td>
</tr>
</tbody>
</table>

Every node with wind as its subtype represents a wind farm with a total of 175 wind turbines per farm.

- **Photovoltaic generators**
  The production of Photovoltaic generators is based on the longitude, latitude, year and month as defined in the configuration file [21]. In addition we also take into account the efficiency of the panels when calculating their actual production output. The photovoltaic generators resemble photovoltaic panels with central inverters, covering approximately 55 hectares. Given this configuration the maximum power generation capacity of the Photovoltaic generators is 55MW. We assume an efficiency of 15% for the panels.

- **Consumers**
  Nodes with the consumer type draw power from the grid and are thus sink nodes. Because we are only concerned with the transmission grid consumer nodes represent distribution grids to which factories, households, etc. are connected.

- **Storage**
  The storage nodes simulate theoretical ESS. Depending on their state (charging, discharging, neutral) storage nodes can either act like producers, consumers or not interact with the grid at all. Storage nodes have the following attributes:
  - **State of Charge (SoC)** in MWh
  - **Maximum SoC** in MWh
    The maximum energy capacity of a storage node
  - **Minimum SoC** in MWh
    The minimum State of Charge of a storage node. Discharging beyond this point will reduce the lifespan of the storage node and is therefore not desirable.
-- Charge and Discharge efficiency
  Defined in the configuration file the charge and discharge efficiency impacts the amount of energy going into a storage node. We have assumed an efficiency of 87%, giving us a round trip efficiency of 75%.

-- Power Capacity in MW
  The maximum power flow the storage can handle for charging or discharging.

• Inner nodes
  Inner nodes represent transmission substations of the grid. They connect transmission lines, conventional generators, renewable generators, storages, and consumers to the grid.

3.3 Topological representation of a power grid

[12] represent the electricity grid as a graph therefore it is worthwhile to establish the relevant definitions and concept from graph theory [12]:

Definition 3.3.1 (Network). A network is a collection of points, also called vertices or nodes connected by lines which are also called edges. Vertices belonging to an edge are referred to as end-vertices.

Definition 3.3.2 (Power grid graph). A power grid pair is a pair $(V, E)$ where each element $v_i \in V$ is either a substation, transformer, or producing/consuming unit of a physical power grid. There is an edge $e_{ij} = (v_i, v_j)$ with $e_{ij} \in E$ between two nodes if there is a physical cable directly connecting the elements represented by $v_i$ and $v_j$.

Definition 3.3.3 (Undirected graph). An undirected graph is a network in which each edge has no orientation: $\forall (v_i, v_j) \in E$ then $(v_j, v_i) \in E$.

Definition 3.3.4 (Flow). A flow over a graph is a real function $f : V \times V \leftarrow \mathbb{R}$ such that:

$$-c(u, v) \geq f(u, v) \geq c(u, v) \quad (3.3a)$$

$$f(u, v) = -f(v, u) \quad (3.3b)$$

$$\sum_{u, v \in E} f(u, v) = \sum_{v, z \in E} f(v, z) \quad (3.3c)$$

For each vertex $v \in V$.

Equation (3.3a) is the flow constraint: The flow cannot exceed the capacity of the edge.

Equation (3.3b) is the skew symmetry: Flow from $v$ to $u$ must be opposite to the flow from $u$ to $v$.

Equation (3.3c) is the flow conversation: The new flow to a node is zero, except for sources which 'produce' power flow, and sinks which 'consume' power flow.
**Definition 3.3.5 (Path and path length).** A path of graph \( G \) is a sub-graph \( P \) of the form:

\[
V(P) = \{v_0, v_1\} \ldots v_l, E(P) = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{l-1}, v_l)\}
\]

Such that \( V(P) \subseteq V \) and \( E(P) \subseteq E \). The vertices \( v_0 \) and \( v_l \) are the end vertices of \( P \) and \( l = |E(P)| \) is the length of \( P \), that is the number of edges that path \( P \) contains.

**Definition 3.3.6 (Shortest path).** Given a graph \( G \), the shortest path from \( v_i \) to \( v_j \) is the path corresponding to the minimum of the set \( \{|P_1|, \{P_2, \ldots, |P_k|\} \} \) containing the lengths of all paths for which \( v_i \) and \( v_j \) are the end vertices.

**Definition 3.3.7 (Betweenness centrality of a node).** The Betweenness centrality of \( C_b(v) \) of a vertex \( v \in V \) is:

\[
C_b(v) = \sum_{s,t} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}
\]

Where \( \sigma_{s,t} \) is the total number of shortest paths from node \( s \) to node \( t \) and \( \sigma_{s,t}(v) \) the number of paths that pass through vertex \( v \).

**Definition 3.3.8 (Utilization index).** Let \( f_e \) be the flow passing through edge \( e \), let \( c_e \) be its capacity. The utilization index \( U(e) \) is:

\[
U(e) = \left| \frac{f_e}{c_e} \right| \cdot 100
\]

\( U(e) \) expressed the percentage of a line’s capacity. In this way the edges can be ranked according to the portion of flow that they carry.

### 3.3.1 Definitions specific to the power grid

In this work we are concerned with the application of graph theory for power grids. Therefore we introduce several important concepts and definitions from [12].

**Definition 3.3.9 (Weighted power grid graph).** A weighted power grid graph is defined as \( G_w(V, E) \) with an additional function \( w : E \rightarrow \mathbb{R} \) associating a real number to an edge representing the reactance of the cable (per unit).

The nodes that are used in \( G_w(V, E) \) mostly overlap with the definitions from Fiorini et al, for the purposes of this work however some of these definitions have been split up or renamed. For example; generators to conventional generators and renewable generators. The following nodes types are present in \( G_w(V, E) \):

- **Wind generators**: \( wg_i \in W \) with \( W \subset V \), where \( W \) is the set of all wind-farm nodes.
- **Photovoltaic generators**: \( pvg_i \in P \) with \( P \subset V \), where \( P \) is the set of all photovoltaic nodes.
- **Conventional generators**: \( cg_i \in G \) with \( G \subset V \), where \( G \) is the set of all conventional generators.
• Consumers: \( c_i \in C \) with \( C \subset V \), where \( C \) is the set of consumer nodes.

• Storages: \( s_i \in S \) with \( S \subset V \), where \( S \) is the set of storage nodes.

• bus bars \( i \in I \) with \( I = V \setminus \{ P \cup W \cup G \} \cup C \cup S \), Where \( I \) the set of inner nodes.

The phase angle property \( \theta \) is assigned to all nodes.

Following the work done by [12], \( W \cup P \cup G \) contains all source nodes. A source node injects power into the grid which has to be transported over edges in the graph to consumer nodes. Consumer nodes in set \( C \) can be considered sink nodes and draw power from the grid. Storage nodes from set \( S \) can either act as sinks, when they are charging or as sources when they are discharging. A node is considered an inner node when two or more lines are connected to it. The lines connected to such a node are weighted with reactance.

Edges in \( G_{\text{in}}(V,E) \) are divided into two categories; real edges, and virtual edges. We consider an edge real if that edge represent the physical properties of a real physical transmission line. The reactance per unit and capacity of a real edge is set to equal to capacity and reactance of the physical transmission line. A edge is considered a virtual edge if it connects a node contained in \( G \cup C \cup S \cup P \cup W \) to the grid, in this case its reactance is set to \( 10^{-4} \) and the capacity is set to equal the maximum load of the consumer, or the maximum generation capacity a generator.

3.4 Different types of graphs

In this work we present three different types of graphs; the modified IEEE-96 bus section 2.4.4 created by [12], a topology based on the Watts-Strogatz model, and a topology based on preferential attachment model as explained by [8]. The modified IEEE-96 bus is presented in section 2.4.4 in the following sections we present the small-world model from [22], and the preferential attachment model from [8]. It should be noted that for both of these models we have used the implementation of [6].

3.4.1 Small world graph

Watts and Strogatz developed a model [22] that allows for the generation of graphs with the small-world property. This model works as follows:

Starting with a regular ring lattice of \( N \) nodes, each \( N \) is connected to its \( K \) nearest neighbor. A vertex and edge is then chosen from the ring in a clockwise manner. This edge is then reconnected with probability \( p \) to an vertex chosen uniformly from the entire ring. This process is then repeated for a full lap of the ring, duplicate edges are not allowed and if they may occur the edge is left alone. This process is then repeated for \( \frac{2\pi}{p} \) laps. Probability \( p \) controls the amount of randomness in the network, if \( p = 0 \) the original ring is unchanged, while \( p = 1 \) will generate a network in which all edges are randomly assigned. Watts and Strogatz defined the small-world property as \( L \gtrsim L_{\text{random}} \) and \( C \gtrsim C_{\text{random}} \) where \( L \) is the characteristics path length of the graph we are considering, \( C \)
is the average clustering coefficient. \( L_{\text{random}} \) and \( C_{\text{random}} \) are respectively the characteristic path length and clustering coefficient of a random network with the same number of vertices and the average number of edges per node.

Shown in Fig. 3.1 is the small world topology that was used in this work, in this case with distributed storage near the consumers.

Figure 3.1: Small world topology with distributed storage

### 3.4.2 Preferential attachment graph

We have used the Barabasi–Albert model for generating preferential attachment topologies. This model uses the concept of preferential attachment to assign edges to nodes. This concept is based on the observation that the more important a node is the more likely it is that other nodes are linked to it. This phenomenon is observed by [8] in the way web pages are linked on the world wide web and it was found when examining the collaboration of movie actors. Preferential attachment is said to be scale free and in order to generate these networks Barabasi and Albert present the following model:
We begin with a network with a small number of vertices \( m_0 \), at every time step we add a new vertex with \( m(\leq m_0) \) edges linking vertex \( m \) to vertices already in the graph. The probability that \( m \) is assigned to a node \( i \) is \([8]\):

\[
p_i = \frac{k_i}{\sum_j k_j}
\]

(3.6)

Where \( k_i \) is the degree of node \( i \). After \( t \) time steps this model leads to a random network with \( t + m_0 \) vertices and \( mt \) edges \([8]\). Barabasi and Albert note that this algorithm will lead to scale-free networks.

Shown in fig. 3.2 is the preferential attachment topology that we used in this work.

![Figure 3.2: Preferential attachment topology with distributed storage](image)

3.5 Economical model for the cost of storage

Our approach to calculating the cost of storage is very similar as those used in \([17]\). Let the annualized investment cost per MWh be:

\[
IC_e = c_{en} \frac{r(1 + r)^h}{(1 + r)^h - 1}
\]

(3.7)

And the annualized investment cost per MW:

\[
IC_{pow} = c_{pow} \frac{r(1 + r)^h}{(1 + r)^h - 1}
\]

(3.8)
Where $c_{en}$ is the cost per MWh and $c_{pow}$ is the cost per MW of unit storage. Let $h$ be the equipment lifetime and let $r$ be the annual interest rate. The cost of the storage is then calculated as:

$$Cost_{storage} = SoC_{Max}IC_{en} + chIC_{pow}$$ (3.9)

Where $SoC_{Max}$ is the maximum SoC of the storage in MWh and $ch$ is the rate of (dis)charge in MW. For this work we have assumed an equipment lifetime of 20 years and an interest rate of 5%.

### 3.5.1 Savings from storage

The savings from storage are calculated as:

$$Saving = Cost_{eens,wo} - Cost_{eens,w}$$ (3.10)

Where $Cost_{eens,wo}$ is the cost of EENS of a simulation without storage and $Cost_{eens,w}$ the cost of EENS from a simulation with storage. Equation (3.10) is the saving in cost of EENS caused by the introduction of a storage given a certain size and (dis)charge rate.

The return from storage is calculated as:

$$return_{storage} = \frac{Saving}{Cost_{storage}} \times 100$$ (3.11)

With eq. (3.11) we can determine if a storage unit can pay for itself by reducing the cost of EENS. If the value of eq. (3.11) is above 100 then the Transmission System Operator (TSO) profits from the storage, if the value is 100 then the storage has paid for itself in full. Negative values indicate that the storage cannot pay for itself, this occurs when there is an insufficient reduction in the cost of EENS.
Chapter 4

Concept and Realization

This chapter begins with section 4.1 which describes the overall procedure that takes place when running the simulation program. Section 4.1 is followed by section 4.2 which describes the architecture of the simulation program, including the different input files and configuration files that the program uses. In section 4.2 we present the architectural layout of the simulation program using class diagrams. Finally Section 4.3 describes the architecture of the grid visualization tool and explains how use this tool to create or modify grid layouts.

4.1 General procedure

The procedure for computing the power flow is divided into two distinct phases. A planning phase and a real time operation phase. The following sections describe these phases and what occurs within each phase.

4.1.1 Planning phase

Conceptually the planning phase represents the day ahead planning that is done by the TSO and conventional generator operators. During this phase we conduct the necessary operations in order to plan production for conventional generators, calculate the expected production of renewable generators, and the expected load of consumers. In addition to this we also plan charging the ESSs according to a predefined policy. What follows is a description of what occurs per node type during this phase.

Consumers

The load that a consumer places on the grid is defined by considering:

1. The total hourly load over a 24 hour period.
2. A percentage of the total hourly load, henceforth referred to as expected load.

The total daily load is defined in an input file that is read by the program. This input file describes for each hour (24 in total) what the cumulative load is on the network. That is to say the input file lists for each hour over a 24 hour period
what the total load is from all consumers. The expected load is then computed as a percentage from the total hourly load: \( \text{expectedLoad} = \text{thl}_t \times \text{percon}_i \), where \( \text{thl}_t \) is the total hourly load at time \( t \), and \( \text{percon}_i \) is the consumption of consumer \( i \) as a percentage of the hourly load. Appendix A contains the input file used for this purpose. The percentage that a consumer takes from the total hourly load is given in the grid configuration file.

**Storage**

In the configuration file there are two settings which are used to define a mandatory charging period. The first setting defines the start time of this period; the second defines the end time. Using these settings we have defined a period during the night (from 23:00 to 05:00) during which the storage must charge. During this night period we plan to charge the storage to 50% of their maximum capacity while of course respecting the maximum power flow that the ESS can handle. It should be noted that during the real time phase the storage may be charged beyond this 50% point if there is an excess of renewable energy. The purpose of this planning phase is to ensure that the production of conventional generator is set high enough such that the storage can always be charged to 50% of their Maximum State of Charge regardless of renewable energy production.

**Renewable Generators**

During the planning phase we calculate the expected renewable production. The expected renewable production represents an estimation of the actual renewable production that we find during the real time phase. The expected renewable production is calculated because it introduces uncertainty into the simulation that is inherent to generating power from wind and solar. That is to say when planning the conventional production one cannot say for certain what the weather will be ahead of time and therefore there needs to be a difference between the renewable production during the planning phase and the real time phase.

**Conventional generators**

Following the calculation of the expected renewable production for solar and wind farms, the load from consumers, and the load from ESSs, the production of conventional generators is set to equal the total load on the grid from consumers and ESSs. When planning the production we take into account buffers such that during the real time operation phase the production can always be increased or decreased. These buffers amount to 7.5% below the maximum production capacity and 7.5% above minimum production capacity. When setting the production of conventional generators we take into account the expected renewable production to limit curtailment. That is to say the production of conventional generators plus the production of renewable generators should equal the load of consumers and ESSs.

### 4.1.2 Real time phase

Having calculated the expected load for the consumers, the expected renewable production, and expected conventional production, we can proceed to the real
time phase. This section explains what occurs during this phase for each of the different node types.

**Consumers**

There is of course a difference between the [Expected load] and the real load one encounters during real time operation of the grid. We have therefore defined the [Real load] as the load that is placed on the grid during real time operation. This load differs from the [Expected load] because we calculate a cumulative load error which is added to the expected load. Using this cumulative load error allows us to increase or decrease the load on the grid with a degree of randomness due to the use of MCDs when calculating the real load. This approach creates different scenarios in which the consumption either increases or decreases.

**Conventional Generators**

For conventional generators we have included a probability of generator failure. Therefore the first step during the real time operations is to check if the generator has failed. As the [Real load] is different from the [Expected load] for which we planned our production we must now adjust the production of conventional generators such that the production from renewable sources and conventional generators equals the [Real load].

In order to adjust the production we must first introduce the concept of offers. Offers are planned production increase or decrease that the operators of a conventional power generator submit to the [TSO]. The [TSO] can use these offers to adjust the production of conventional generators in order to balance the production and load. For each conventional generator subtype we have declared four offers. Two increase production and two decrease production. As a rule the further away an offer is from the planned production the more expensive the offer is. The order in which offers are accepted depends solely on the price of the offer. Therefore conventional generators with the cheapest offers are adjusted first. There exists a special case in which production has been lowered in order to meet a very low load demand and in which over production still exists. For this case it is possible as a last resort to turn small conventional generators off. Small meaning a generator with a maximum generation capacity no higher than 60MW.

**Renewable generators**

For all of the renewable generators we redo the process of setting there production just like we did during the planning phase. Because there is an element of randomness due to the MCDs the real renewable production will differ from the expected renewable production. The production that is calculated during this phase is the maximum available production. When calculating the flow on the grid the actual power injected into the grid by renewable generators can be below or equal to the maximum renewable production.
Storage

During the planning phase we have planned to charge the ESS to 50% of their maximum energy capacity. In the real time phase this plan is implemented and thus we charge the ESS during the night period. If we find that the real renewable production is higher than the expected renewable production we charge the ESS past 50% of their maximum storage capacity in order to minimize the curtailment. During the day period the TSO can discharge or charge the storage as needed in order to balance the grid.

4.2 Program Architecture

This section presents the overall flow of the program, the input files that the program uses and the architecture of the simulation program using class diagrams. In addition we also list the tools and technologies that we have used for this work.

4.2.1 Used Technologies

Java 1.8 was used as the programming language; hence the program requires Java 1.8.0_101 in order to run. Because the balance between production and consumption has been expressed as a linear programming problem we use the GLPK v4.52 library to solve the power flow computations. Eclipse Neon Release 4.6.0 and Mars Release 4.5.1 was used as the development environment. In order to compute the different metrics, and to visualize and edit the grid GraphStream version 1.3 was used. Github was used for version control and the simulation program and grid visualization tool are available on and . Maven was used for both of these programs in order to manage the dependencies.

4.2.2 Program flow

In section 4.1.1 we have described the representation of an electricity grid as a graph. Section 4.1.1 also describes the different node properties and the action performed by these nodes during the two different phases of operation. Before going into depth of how the simulation program works it is of course helpful to have a global overview of the program flow. Shown in fig. 4.1 is such an overview. What follows is a description of the different elements shown in fig. 4.1.

Read input files

Before beginning with the planning phase the program must of course read the input files. The program requires 8 input files:

- Expected Load spring.csv
  Contains values that describe the total hourly load over a 24 hour period during spring.

- Expected Load summer.csv
  Contains values that describe the total hourly load over a 24 hour period during summer.
• Expected Load fall.csv
  Contains values that describe the total hourly load over a 24 hour period during fall.

• Expected Load winter.csv
  Contains values that describe the total hourly load over a 24 hour period during winter.

• network.csv
  This file contains all the information required by the program in order to build a topology. It contains declarations of all the nodes and their attributes, the declarations of the edges and the attributes of edges. This default file represents the modified IEEE-96 bus.

• modelday.mod and modelnight.mod
  These files contain the model used by the linear program to compute the power flow. Because the rules governing the use of storage during the night period and day period are different we have created different models to reflect this. The modelday.mod file allows the linear program to use the storage as required to balance the grid, of course staying within the limits of the storage (max/min capacity, etc). The modelnight.mod will always force the linear program to charge the storage.

• application.conf
  In order to make the program more flexible we have placed many of the settings the program uses in this file. Should the user desire he can easily change these settings. Among them we have settings for parameters of the distributions that were used for the MCDs. Additionally we have settings for the price of the offers, wind and solar generator settings, and storage settings.

The exact contents of these files are available in appendix A.

**Planning phase**

Once all input data has been read and parsed we can begin the planning phase. In this phase we calculate the expected load, set the required production of conventional generators to meet this load plus the load incurred from charging the ESS to 50% of their SoC. In addition to this we also compute the expected production of the renewable generators. Following the calculation of the expected production and expected load we can proceed to the real time phase.

**Real time phase**

The first step in the real time phase is to calculate the real load that consumers place on the grid. This is done calculating a cumulative load error and adding it to the load of a consumer. The resulting value is referred to as real load and differs slightly from the expected load. The real renewable production is then computed by repeating the same process that was used for computation of expected renewable production. Because the process of calculating expected
renewable production uses MCDs, repeating the process will change the maximum available production. At this point we have the real load from consumers, ESSs, and the maximum available production of renewable generators. We can therefore begin adjusting our conventional generators by increasing or decreasing their production using the offers located in application.conf. If we have overproduction caused by conventional generators we can as a last resort turn off certain generators that have a maximum generation capacity of 60MW.

**Linear program**

At this point the program creates an input file for the linear program. This input file is written in the “GNU modeling language”. The load flow is then computed by GLPK [4] by executing a system call to GLPK from Java. The linear program will attempt to minimize the objective function (i.e. minimize cost of EENS and curtailment) by adjusting the amount of power drawn from renewable generators and ESS. Once the linear program has found a solution to minimizing the objective function it creates an output file containing the flow on all the lines. Because the linear program in part sets the state of our graph by adjusting the power drawn from renewable generators and ESS we have to update the representation of the grid in java. Once the representation of the grid in the java program has been updated the program creates output files that can be used for analysis and visualization.

**Hourly iterations, seasonal iterations, and convergence**

The process described in this section is repeated for each hour of a 24 hour period. Following execution of the linear program we check the convergence;

\[
conv = |eens_{avg} - eens_s| \begin{cases} 
1 & \text{if } conv > conv_{config} \\
0 & \text{otherwise}
\end{cases}
\]  

(4.1)

Where \(eens_{avg}\) is the average EENS and \(eens_s\) is the EENS of the last simulation, and \(conv_{config}\) is set by the user in application.conf. If the simulation achieves convergence we move on to the next season. The above procedure is done for winter, fall, summer, and spring using different total hourly loads for each season; the season also impacts the production of solar farms but not the production of wind farms.
Figure 4.1: Flow of the program
4.2.3 Class diagrams

The program is divided into 6 different java packages. For each of these java packages a class diagram has been made. What the different classes within these package do will be explained in this section. A high level overview of how these classes relate to one another is shown in fig. 4.2

![Figure 4.2: Class diagram of the program](image)

Model package

The model package holds classes that resemble different objects that we find on a electricity grid. In fig. 4.3 we see the contents of the model package. The classes contained in this package allow us to create a graph representing a power grid as specified by definition 3.3.2. What follows is a description of these classes and their roles within the program:

- **Generator**
  The generator class is the super class of ConventionalGenerator and RenewableGenerator. This approach was taken because ConventionalGenerators and RenewableGenerators share many of the same attributes. The subtype of a generator is defined by the enum GENERATOR_TYPE.

- **ConventionalGenerator**
  This class resembles the conventional generator objects such as coal power plants.
• GENERATOR_TYPE
  Enum type used to define the subtype of a generator. The subtypes are; wind, solar, coal, nuclear and hydroelectric.

• RenewableGenerator
  Renewable generators can either resemble photovoltaic solar panels, or wind turbines. The subtype is defined by GENERATOR_TYPE.

• Consumer
  Consumer nodes represent distribution grids that draw power from the transmission grid.

• Energy Storage System
  In this case the ESSs represent a theoretical storage without spin down or spin up times. Since ESSs can produce power by discharging or consume it by charging we have to keep track of its state. This is done by using the StorageStatus enum.

• StorageStatus
  Used to indicate the state of an ESS.

• Offer
  Every conventional generator has a total of four offers attached to it two to increase production and two to decrease production. The order in which the production of the conventional generators is adjusted dependents on the price of their offers. Generators with the cheapest offer are adjusted first. As such one can say that generators are sorted in decreasing order according the price of an offer.
Figure 4.3: Class diagram of the model package

Graph package

The classes within the graph package allow for the construction of a graph as specified by definition 3.3.1. The exact relationships of the classes are shown in fig. 4.2.
• **Node**
  The Node class is a super class for the different kind of nodes in the graph.

• **InnerNode**
  InnerNodes are a subclass of Node and resemble substations and busbars of the transmission grid.

• **Graph**
  The graph class is used to hold the nodes and edges objects in such a way that they resemble a graph. It also contains attributes that are important to the operation the grid/simulation such as the amount of nodes, amount of edges, cost of curtailment, etc. The Graph class contains a method called `printGraph()`. This method is called after each simulation step and writes the current state of the graph to the file system in DGS format. These .dgs files can be used for visualization purposes using the Grid Visualization Tool.

• **Edge**
  Edges connect the nodes in the graph and have two attributes: Capacity, which is the maximum amount of power flow that the edge can handle and reactance, which is an attribute used during the calculation of the load flow to make it more realistic. Finally we have the flow attribute which is used to keep track of the amount of power flowing over an edge.
FileHandler package

The classes within the FileHandler package are used to load the input for the program. The FileHandler package also has classes that create the output files for analysis or visualization purposes.

- OutputFileHandler
  This class is used to place files created by the linear program in folders corresponding to their simulation, and season. In addition this class creates several output files that can be used for analysis, e.g. flow from/to ESS, daily EENS etc.

- DataModelPrint
The DataModelPrint class creates input files required by the linear program so that it can compute the power flow.

- **Parser**
  The Parser class is responsible for reading the input file that describes the grid topology and the properties of its edges and nodes. This includes the network configuration file, total hourly load files as described in section 4.2.2.

![Class diagram of the fileHandler package](image)

**Simulation package**

- **SimulationStateInitializer**
  This class is responsible for setting the initial state of a graph. It does so by reading the topology configuration using the Parser class and the configuration file using the Configuration library. This class initializes the graphs for every timestep, preparing them for the planning and real-time phases.

- **EENSHandler**
  The EENSHandler class is used to calculate the EENS.

- **GridBalancer**
  The GridBalancer is responsible for performing operations in order to balance the production and load. This includes implementing the policy of charging storages during the night period, and adjusting the production using the offers.

- **ProductionLoadHandler**
The ProductionLoadHandler class is responsible for calculating the expected load of consumers and expected production of generators. In addition to this the ProductionLoadHandler class is also used to plan charging the storage during the night period.

- **StorageHandler**
  The StorageHandler class contains the logic required to plan charging the storage during the night period. Additionally it is also responsible for implementing those plans, and charging the storage beyond 50% of their energy capacity if there is an excess of renewable energy.

- **MonteCarloHelper**
  This class contains methods that are responsible for performing different [MCDs]. This includes draws from the following distributions: Normal, Weibull, Uniform, and Gamma.

- **SimulationMonteCarloDraws**
  This class is responsible for setting the failure state of conventional generators and setting the expected and real production of renewable generators.

![Class diagram of the simulation package](image.png)

Figure 4.6: Class diagram of the simulation package
4.2.4 Usage instruction

The simulation can be started by opening a terminal and navigating to the location of the Simulation.jar file. Once there one can start the simulation by using the command java -jar Simulation.jar.

4.3 Grid Visualization Tool

The Grid Visualization tool was developed to: a) Analyze the state of the grid at specific time steps and b) Generate, modify, or alter grid topologies. Section 4.3.1 details the architecture of the tool and instructions in order to use it.

4.3.1 Architecture

In Fig. 4.7 we see the overall architecture of the Grid Visualization Tool. The tool is built using the GraphStream library. GraphStream is a “Java library for the modeling and analysis of dynamic graphs.” In Fig. 4.7 we see an overview of the Grid Visualization Tool classes. The following sections give a description of these classes.

GUI

The GUI class is responsible for creating and managing the graphical user interface. It holds all of the variables, and logic that is required for user interaction. The Java Enumeration VIEW_MODE connected to this class is used to switch between two different modes of the tool depending on which tab is selected by

Figure 4.7: Class overview of the Grid Visualization tool
the user as shown in fig. 4.12 and fig. 4.13. The KEY_DOWN Enumeration is used to communicate key presses to a number of different methods.

Figure 4.8: GUI class

GraphGenerator

The graph generator currently allows for the generation of Small World graphs and preferential attachment graphs. The overall procedure for the generation of these graphs is the same, except for the way nodes are attached. For the small world graph we begin by creating a small world network as specified by \[22\] and implemented by [6]. The initial network exists solely of inner nodes and edges with a reactance higher than 0.0001. When generating our initial graph \( n \) is set to equal the amount of inner nodes that we want, \( k = 2 \), and \( \beta = 0.5 \). Following this we assign the outer nodes in the following manner: Begin by selecting a random inner node with a uniform probability, to this inner node attach an outer node and assign an edge between them with a capacity that equals the maximum generation capacity or consumption of the outer node. This process is repeated until all outer nodes have been assigned to random inner nodes. At this point the user is left with edges that have not been assigned yet; these edges can then be assigned manually in order to ensure that the resulting grid has the small world property.
The generation of preferential attachment graphs works similarly. First generate a preferential attachment graphs using GraphStream, then we assign the outer nodes by selecting random inner nodes and assigning an edge with enough capacity in order to transport the flow to or from the outer node.

![Figure 4.9: GraphGenerator class](http://graphstream-project.org/doc/Generators/Barabasi-Albert-Preferential-Attachment-generator/)

### 4.3.2 GraphLogic

The GraphLogic class is responsible for managing user interaction with a graph. It allows the user to add nodes, remove nodes and assign nodes a type (conventional generator, consumers, etc). This class also allows the user to load in graphs from a grid configuration file, and it allows the user to load in .dgs files in order to visualize the grid on an hour by hour basis.

![Figure 4.10: The GraphLogic class](http://graphstream-project.org/doc/Generators/Barabasi-Albert-Preferential-Attachment-generator/)
**GraphMetrics**

The GraphMetrics class is responsible for calculating different metrics of graphs. It allows for the computation of the small world property, characteristics path length, betweenness centrality as defined in section 2.5 and definition 3.1.2. The betweenness centrality is calculated using the GraphStream library. The GraphStream library has several useful metrics available through its Toolkit class, examples of this include a method to calculate the diameter and Clustering coefficient.

![GraphMetrics class](image)

**Figure 4.11: The GraphMetrics class**

### 4.3.3 Usage Instruction

The visualization tool has been split into two modes; inspection and generation. The user can select which mode he wants to use by selecting the respective tab in the upper right corner as shown in fig. 4.12 and fig. 4.13. The layout of the nodes is determined by a force-repulsion algorithm that comes with the GraphStream library, the user can toggle this force-repulsion algorithm by pressing the `spacebar`. In addition if the user somehow loses track of the position of the graph, he can recenter the view by pressing `shift+r`. In order to easily identify nodes, the nodes have been colored according to their types:

- **Yellow nodes gray outline**
  Consumer nodes with satisfied loads. That is to say the flow on the line is equal to the load of the consumer.

- **Yellow nodes with red outline**
  Consumer nodes with EENS. That is to say the flow on the line is smaller than the load of the consumer.

- **Green nodes**
  Renewable generators either wind or photovoltaic.

- **Gray nodes green outline**
  Conventional generators of different subtypes.

- **Gray nodes red outline**
  Conventional generators that have failed and cannot produce.

• Black nodes gray outline
  Inner nodes.

• Blue nodes gray outline
  Storage nodes.

**Inspection mode**

In the inspection mode the user can select a folder that contains a set of Graphstates files, denoted by the .dgs extension. In this mode the user can select a node by left clicking on it which will make the program display information related to that node and the lines connected to the node. The information that is shown depends on the type of node selected; e.g. if one selects a consumer node the program shows the load the consumer has and the flow going into that consumer via an edge. The numbers located beneath the different nodes indicate the node Ids as set by the grid configuration file. In order to ease the identification of congested lines the edges are given three different colors depending on the flow and its capacity. The user can inspect different hours by selecting the corresponding .dgs from the dropdown list. The arrows seen on the edges indicate in which way the network configuration connects the nodes and are not indicative of direction of the power flow.

- **Green**
  Flow on the lines is lower than 1/3 of the line capacity.

- **Orange**
  Flow on the line falls between 2/3 and 3/3 of the line capacity.

- **Red**
  Flow on the line exceeds the capacity.

![Figure 4.12: Grid Visualization Tool Inspection](image)
Press and hold | Action
---|---
A | Connects the node clicked on with the 1st edge from the edge list to the 1st node in the node list.
D | Remove the node clicked on and re-add it to the node list, when removing a node edges connected to that node are removed and re-added to the edge list if their degree is 2.
E | Adds the first edge from the node list between the node that was clicked on first and the node that was clicked on second.

Table 4.1: Key layout of the Grid visualization tool

**Generation mode**

In the generation mode the user can create new graphs, and edit existing ones. In order to generate small world graphs, or preferential attachment graphs, the user can click on the button marked as such and select an existing network file. Using the nodes and edges found in this file, a corresponding topology will then be created as described in section 4.3.1. It should be noted that as the assignment of edges between inner nodes is random this process does not guarantee that the topology that is created is viable for the simulation. In certain cases it is therefore required that the capacity of the grid is increased, or that edges are removed or added in order to make the grid viable for the simulation. The task of guaranteeing that a generated topology is viable for the simulation falls outside the scope of this work. When the save button is pressed to program saves the graph as graph.mod in the root folder. This file can then be used by the simulation program. Shown in table 4.1 is the key layout that allows the user to interact with the graph when in generation mode.
Figure 4.13: Grid Visualization Tool following the initial generation of a small world graph
Chapter 5

Mathematical formulation

In this chapter we present the mathematical models used by our simulation. We begin by presenting the models for the different node types and the transmission lines in section 5.1. Following this we also present the linear program and its model in section 5.2.

5.1 The models

In this section we present the mathematical models that were used for the simulation of wind turbines, photovoltaic panels, conventional generators, consumers, ESSs and transmission lines.

5.1.1 Wind generators

The power generated from wind turbines is calculated by eq. (5.1). The values for vCutIn, VRated are described in section 3.2.2. Wind speed is represented by $ws$. The value of $ws$ is drawn from a weibull distribution with the following parameters: shape = 1.6, scale = 8. Let $P_{w,\text{rated}}$ be the rated power of the turbines. For eq. (5.1) we have assumed a cubic dependency between the relation of generated power of a wind turbines and the wind speed, similar to the approach of [18].

$$P_w = P_{w,\text{rated}} \frac{w^3 - v_{\text{CutIn}}^3}{v_{\text{Rated}}^3 - v_{\text{CutIn}}^3} \begin{cases} 
\text{if } ws \leq V_{\text{CutIn}} & | P_w = 0 \\
\text{if } ws \geq V_{\text{CutOff}} & | P_w = 0 \\
\text{if } ws \leq v_{\text{Rated}} \land ws \leq V_{\text{CutOff}} & | P_w = P_{\text{Rated}}
\end{cases} \tag{5.1}$$

5.1.2 Photovoltaic generators

The approach for the PVS is based on [21], [19], and [19]. [21] describes the output of PVS as:

$$P_p = A_c \eta I_b \tag{5.2}$$

Where $A_c$ is the array surface in $m^2$, $\eta$ is the panel efficiency and $I_b$ is the irradiance on a surface with inclination $\beta$ to the horizontal plane. Since the
irradiance $I_\beta$ is dependent on the latitude and climate conditions (cloud cover) we calculate $I_\beta$ as follows [9].

First we calculate $S_{\text{max}}$ which is the peak solar irradiance.

$$S_{\text{max}} = I_{\text{ETI,max}}M$$  \hfill (5.3)

$I_{\text{ETI,max}}$ is the peak maximum solar radiance to the daily maximum extraterrestrial irradiance. $I_{\text{ETI}}$ is the extraterrestrial irradiance defined as [19]:

$$I_{\text{ETI}} = I_0E_0 \cos(\theta_z) = 1362 \left(1 + 0.033 \cos \left(\frac{2\pi d_n}{365}\right)\right) \cos(\theta_z)$$  \hfill (5.4)

Where $I_0$ is the solar constant in $W/m^2$, $\theta_0$ the eccentricity correction factor $d_n$ is the number of the day in the year and $\cos(\theta_z)$ the zenith angle in degrees. $M$ represents the cloudiness. For our purposes we have used the definition of cloudiness as described by [9]. Therefore $M$ is a MCD from a Gamma distribution with $\lambda = 8.9166$ and $C = 0.0311$. Following this we calculate the $I_\beta$ as:

$$I_\beta = S_{\text{max}} \sin \left(\frac{\pi t - t_{\text{rise}}}{t_{\text{set}} - t_{\text{rise}}}\right)$$  \hfill (5.5)

Where $t$ is current time of day, $t_{\text{rise}}$ the time of sunrise and $t_{\text{set}}$ the time of sundown. Following this $P_p = 0$ if $t \leq t_{\text{rise}} \lor t \geq t_{\text{set}}$ otherwise we calculate eq. (5.5). Since $\theta_z$, $t_{\text{rise}}$ and, $t_{\text{set}}$ are dependent on the location of the PVS we calculate these using the date, latitude, longitude and, \Delta T which is the difference between Terrestrial Time, and Universal Time [14]. For exact details of these calculations we refer to [14].

To add variance to the model [9] uses 4 additional parameters namely $a_1$, $a_2$ as the amount of solar irradiance variation, $b_1$, $b_2$ as the frequency of change for weather and $c_1$, $c_2$ as the rate of change in weather. For our purposes however we omitted these parameters since the MCD performed in eq. (5.3) introduces randomness in the model. The zenith angle, and sun rises/sun down times have been calculated using the solarpositioning library [1]

5.1.3 Conventional generators

Every conventional generator has a probability of failure denoted by:

$$\text{failProb}_g = \frac{1}{\text{mttf}_g}$$  \hfill (5.6)

$$\text{failure}_g = \begin{cases} 1 & \text{if } \text{MCD} \leq \text{failProb}_g \\ 0 & \text{else } \text{MCD} > \text{failProb}_g \end{cases}$$  \hfill (5.7)

Where $\text{failure}_g$ is the failure state of generator $g$, $\text{mttf}_g$ is the MTTF of conventional generator $g$. The MCD in this case is taken from a uniform distribution with $\mu = 0$ and $\sigma = 0.5$. Equation (5.6) and eq. (5.7) are computed at the beginning of every hour. If a generator fails, it is out of order for the remainder

[1] https://github.com/KlausBrunner/solarpositioning
of the day.

Every conventional generator has a minimum power generation limit and a maximum power generation limit; \( P_{\text{min}} \) and \( P_{\text{max}} \). \( P_{\text{g}} \) indicates the power output of conventional generator \( g \). Therefore we have the following rule that is used during the planning phase:

\[
(P_{\text{min}} + (P_{\text{max}} \cdot 0.075)) \leq P_{\text{g}} \leq (P_{\text{max}} - (P_{\text{max}} \cdot 0.075))
\]  

(5.8)

During the planning phase production is planned using a 7.5% buffer with respect to the maximum and minimum output of a conventional generator, as shown in eq. (5.8). During the real time phase we use eq. (5.9) such that the full production capacity of a generator can be used.

\[
P_{\text{min}} \leq P_{\text{g}} \leq P_{\text{max}}
\]  

(5.9)

The decision to increase or decrease production depends on the balance of production and consumption therefore we have:

\[
\Delta P_{t} = P_{\text{real}} - L_{\text{real}}
\]

\[
\begin{cases} 
\text{if } \Delta P_{t} < 0, \text{ over production} \\
\text{if } \Delta P_{t} > 0, \text{ under production} \\
\text{if } \Delta P_{t} = 0, \text{ balanced}
\end{cases}
\]  

(5.10)

In eq. (5.10) \( \Delta P_{t} \) denotes the balance between production and load at time \( t \). \( P_{\text{real}} \) is the sum of all power being injected into the grid at time \( t \). \( L_{\text{real}} \) is the sum of power being consumed at time \( t \). Using it we attempt to balance production according to what the current consumption is. In order to decrease or increase production of conventional generators offers are used as described in section [4.1.2](#). Mathematically we minimize the cost to the TSO therefore:

\[
\min \{ f(x) \mid x \in I \lor D \}
\]  

(5.11)

Where \( f(x) \) is the cost incurred by the TSO when changing the production. \( I \) is a set containing the prices for production increases and \( D \) is the set containing the prices for production decreases. Following eq. (5.11) we increase/decrease production as denoted by eq. (5.12) and eq. (5.13).

For increasing:

\[
P_{g,t} = P_{g,t-1} + P_{o,g,x}
\]  

(5.12)

For decreasing:

\[
P_{g,t} = P_{g,t-1} - P_{o,g,x}
\]  

(5.13)

Where \( P_{g,t} \) is the production of conventional generator \( g \) at time \( t \) and \( P_{o,g,x} \) is the increase/decrease in production from offer \( x \) as found by eq. (5.11). \( P_{g,t-1} \) is the production of conventional generator \( g \) at time \( t - 1 \).

### 5.1.4 Consumers

During the planning phase consumers have an expected load defined by the input files. This load is expressed as a percentage of the total hourly daily load for each individual consumer node. For the real time phase we compute the cumulative load error of consumers hence the load becomes:
\[
    l_{i,t} = le_{i,t} + cumle_{i,t}
\]
\[
    ler_{i,t} = le_{i,t} + MCD
\]
\[
    cumle_{i,t} = \sum_{j=1}^{t} ler_{i,j}
\]

Where \( l_{i,t} \) is the load of consumer \( i \) at time \( t \), \( le_{i,t} \) is the expected load from the planning phase, \( ler_{i,t} \) is the load error. The MCD in eq. (5.15) is drawn from a normal distribution with \( \mu = 0 \) and \( \sigma = 0.05 \). Equation (5.16) is the cumulative load error and it is only performed when \( t \geq 0 \) otherwise we compute eq. (5.15).

### 5.1.5 Storage

During the planning phase the total load placed on the grid is influenced by the ESSs because we plan to charge the storage to 50\% of their maximum SoC. During the real time phase the storage can charge beyond 50\% SoC max to prevent curtailment of renewable generators.

Therefore during the real time phase we calculate the excess in renewable power at time \( t \):

\[
    P_{rew,t} = \left( \sum_{w \in W} P_{w,t} + \sum_{p \in P} P_{p,t} \right) - \left( \sum_{c \in C} l_{c,t} - \sum_{g \in G} P_{g} \right)
\]

Then we minimize curtailment of renewable power by computing for every storage node \( s_i \) at time \( t \):

\[
    SoC_{s_i,t} = SoC_{s_i,t} + \text{flow}_{s_i}
\]

\[
    \begin{cases} 
    \text{flow}_{s_i} = P_{rew,t} \cdot \text{chEfficiency} \\
    \text{flow}_{s_i} = \text{flowlimit}_{s_i} \cdot \text{chEfficiency} 
    \end{cases}
\]

Where \( SoC_{s_i} \) is the SoC of storage \( s_i \) at time \( t \) in MWh. \( \text{flow}_{s_i} \) is power entering storage \( s_i \) in MW.

The \text{flowlimit} is defined by:

\[
    \text{flowlimit}_{s_i} = \text{chEfficiency}_{s_i} \cdot \text{chMax}_{s_i}
\]

\[
    \text{flowlimit}_{s_i} = \text{dchEfficiency}_{s_i} \cdot \text{chMax}_{s_i}
\]

Where eq. (5.19a) is the \text{flowlimit} during charging and eq. (5.19b) the \text{flowlimit} during discharging. \( \text{chEfficiency}_{s_i} \) and \( \text{dchEfficiency}_{s_i} \) are the charge and discharge efficiency of storage \( s_i \). \( \text{chMax}_{s_i} \) is the rate of (dis)charge without considering losses due to efficiency. Using eq. (5.18) we minimize the curtailment of renewable energy since the excess of renewable energy \( P_{rew,t} \) is used to charge the storage past 50\% of the maximum SoC. Of course we do not exceed the maximum limit of the SoC.
5.1.6 Transmission lines

Following definition 3.3.9 we defined a generic edge line$_{i,j}$ as a line connecting node $i$ to node $j$:

$$line_{i,j} = \frac{\theta_i - \theta_j}{w_{i,j}} \ast m\text{Factor}$$

(5.20)

Where $w_{i,j}$ is the reactance per unit and $\theta_n$ is the phase angle of node $n$. $m\text{Factor}$ is a multiplication factor required in order to compute the flow on a line in MW as specified by [12]. We consider an edge real if $w_{i,j} > 0.0001$ otherwise we consider the edge to be virtual edge.

5.2 Linear program

In order to compute the power flow we have created a linear program and a corresponding model with an objective function and constraints. The linear program and its corresponding model are presented in section 5.2. For this section we will use the sets as they have been defined in section 3.3.1. To remind the reader:

- Wind generators: $w_{gi} \in W$ with $W \subset V$, Where $W$ is the set of all wind-farm nodes and $V$ the set of all nodes.
- Photo voltaic generators: $p_{gi} \in P$ with $P \subset V$, where $P$ is the set of all photovoltaic nodes.
- Conventional generators: $c_{gi} \in G$ with $G \subset V$, where $G$ is the set of all conventional generators.
- Consumers: $c_i \in C$ with $C \subset V$, where $C$ is the set of consumer nodes.
- Storages: $s_i \in S$ with $S \subset V$, where $S$ is the set of storage nodes
- Bus bars $in_i \in I$ with $I = V \cup P \cup W \cup G \cup C \cup S$, inner nodes.

Let $f(i,j)$ be the power flow between node $i$ and $j$.

5.2.1 The model

First let us begin by defining the objective function, which seeks to minimize the cost incurred by a TSO by minimizing load shedding and curtailment of renewable power sources.

$$\sum_{t=0}^{24} \left[ \min \left( \sum_{v_i \in P \cup W} cost\text{curt} \left( \sum_{v_j \in V} \left( \frac{\theta_i - \theta_j}{w_{i,j}} \ast m\text{Factor} \right) \right) + \sum_{v_i \in C} cost\text{ls} \left( load_i - \sum_{v_j \in V} \left( \frac{\theta_i - \theta_j}{w_{i,j}} m\text{factor} \right) \right) \right] \right]$$

(5.21)

Where $w_{i,j}$ is the reactance of line $ij$, $\theta_i$ the phase angle of node $i$, $cost\text{curt}$ is the curtailment cost of renewable generators, and $cost\text{ls}$ is cost of load shedding. The objective function must fulfill the following constraints:

47
• Slack bus as defined by [12]:
\[ \theta_{46} = 0 \]

• Maximum and minimum flow on a line where \( f_{i,j} \) is the power transfer between node \( i \) and \( j \). \( \forall (i,j) \in V \):
\[ -\text{capacity}_{i,j} \leq f(i,j) \leq \text{capacity}_{i,j} \]

Where \( \text{capacity}_{i,j} \) is the transfer capacity of the line connecting node \( i \) to node \( j \).

• Flow conservation for inner nodes \( \forall n \in I \) [12]:
\[ \sum_{(u,n) \in E} f(u,n) = \sum_{(n,z) \in E} f(n,z) \]

• Constraint for maximum and minimum renewable production \( \forall rg \in P \cup W \)
\[ 0 \leq P_{rg} \leq P_{\text{max}_{rg}} \]

Where \( P_{rg} \) is the production of renewable generator \( rg \) and \( P_{\text{max}_{rg}} \) is maximum available production of renewable generator \( rg \).

• Constraint for conventional production \( \forall g \in G \forall j \in I \):
\[ f(g,j) = P_g \]

Where \( P_g \) is the production of conventional generator \( i \), defined using load balancing and offers.

• Constraint for the load of a consumer \( \forall c \in C \forall i \in I \)
\[ 0 \leq f(i,c) \leq l_c \]

Where \( l_c \) is the load of consumer \( c \) defined using the cumulative load error. The flow going into a consumer node \( f_{i,c} \) can be lower than the load, resulting in EENS.

• Charging during the night period \( \forall s \in S \forall i \in I \):
\[ f(i,s) = \text{charge}_s \]

Where \( \text{charge}_s \) is amount of power going into storage \( s \) defined by our night charging policy for storage node \( s \).

• Constraint storage during day period for charging \( \forall s \in S \forall i \in I \):
\[ \text{discharge}_s \leq f(i,s) \leq \text{charge}_s \]

Where \( \text{discharge}_s \) is the maximum available discharge rate, and \( \text{charge}_s \) the maximum available charge rate.

• Production and load equality:
\[ \sum_{i \in W \cup P \cup G \cup S} \text{line}_{i,j} = \sum_{j \in I} \text{line}_{j,i} \]

• Phase angle constraint from [12]
\[ -\frac{\pi}{2} \leq \theta_{v,j} \leq \frac{\pi}{2} \forall v \in V \setminus s: \text{slack bus} \theta_{s,j} = 0 \forall s: \text{slack bus}. \]
Chapter 6

Results

This chapter reports the results that were achieved using the simulation as described in Chapter 4 and Chapter 5. Section 6.1 describes the procedure used for the sizing of the ESSs and achieved results. In section 6.2, we describe the procedures used for the siting of distributed and centralized ESSs. In section 6.2, we report the results achieved using these siting policies. Section 6.3 introduces several metrics for each topology in order to differentiate them.

6.1 Storage Sizing

In order to find the optimal size of the ESSs, we used the modified IEEE-96 bus shown in Fig. 6.1 with unconstrained lines and a single storage node. We computed the EENS using different sizing profiles as shown in Table 6.1. Due to the wide spread use of renewable energy generation and its impact on energy grids, the Maximum State of Charge was set as a percentage of the total renewable generation capacity available to the grid (5910MW). Sizing profile 0 represents the grid without storage, the subsequent profiles increase the SoC by 5% of the total renewable generation capacity. In example, profile 1: 5% of 5910 = 295MWh, profile 2: 10% of 5910 = 591MWh, etc. The ch represents the (dis)charge rate of the storage and has been set to 15% of the Maximum State of Charge and 30% of the Maximum State of Charge. The economical model used to calculate the cost of storage, the savings by storage and the return from storage's are reported in section 3.5.
Figure 6.1: IEEE 96 modified grid with distributed storage

Shown in fig. 6.2 to fig. 6.4 is the cost of storage, cost of EENS and the savings achieved using the sizing profiles from table 6.1. In table 6.2 we see the returns from price profile 1 (least expensive). Shown in table 6.3 are the returns from storages using price profile 2. And finally in table 6.4 we see the returns from storage using price profile 3 (most expensive). Examining the difference between 15% ch and 30% ch we can say that having a higher (dis)charge rate does not necessarily lead to a lower EENS in our simulation.

<table>
<thead>
<tr>
<th>Sizing profile #</th>
<th>Maximum SoC in MWh</th>
<th>15% ch in MW</th>
<th>30% ch in MW</th>
<th>Savings 15% ch</th>
<th>Savings 30% ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>295</td>
<td>44</td>
<td>88</td>
<td>€420,838,437</td>
<td>€84,011,547</td>
</tr>
<tr>
<td>2</td>
<td>591</td>
<td>88</td>
<td>117</td>
<td>€743,859,662</td>
<td>€848,783,927</td>
</tr>
<tr>
<td>3</td>
<td>886</td>
<td>132</td>
<td>265</td>
<td>€738,861,316</td>
<td>€1,268,002,221</td>
</tr>
<tr>
<td>4</td>
<td>1182</td>
<td>177</td>
<td>334</td>
<td>€725,882,2350</td>
<td>€1,102,790,440</td>
</tr>
<tr>
<td>5</td>
<td>1477</td>
<td>221</td>
<td>443</td>
<td>€1,164,808,915</td>
<td>€1,056,906,005</td>
</tr>
<tr>
<td>6</td>
<td>1773</td>
<td>265</td>
<td>531</td>
<td>€793,194,490</td>
<td>€1,734,284,752</td>
</tr>
<tr>
<td>7</td>
<td>2068</td>
<td>310</td>
<td>620</td>
<td>€1,082,928,135</td>
<td>€1,584,059,886</td>
</tr>
<tr>
<td>8</td>
<td>2364</td>
<td>354</td>
<td>709</td>
<td>€107,760,874</td>
<td>€1,222,711,476</td>
</tr>
<tr>
<td>9</td>
<td>2659</td>
<td>398</td>
<td>797</td>
<td>€1,308,644,353</td>
<td>€1,189,927,614</td>
</tr>
<tr>
<td>10</td>
<td>295</td>
<td>443</td>
<td>886</td>
<td>€1,271,192,600</td>
<td>€1,027,727,931</td>
</tr>
</tbody>
</table>

Table 6.1: Sizing profiles of Storage and their Savings
<table>
<thead>
<tr>
<th>Sizing profile #</th>
<th>Return storage 15% ch</th>
<th>Return storage 30% ch</th>
<th>Cost storage 15% ch</th>
<th>Cost storage 30% ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>301.6%</td>
<td>-33.6%</td>
<td>€139,500,000</td>
<td>€249,500,000</td>
</tr>
<tr>
<td>2</td>
<td>266.5%</td>
<td>169.1%</td>
<td>€279,100,000</td>
<td>€501,600,000</td>
</tr>
<tr>
<td>3</td>
<td>176.5%</td>
<td>168.8%</td>
<td>€418,600,000</td>
<td>€751,100,000</td>
</tr>
<tr>
<td>4</td>
<td>129.4%</td>
<td>109.9%</td>
<td>€500,700,000</td>
<td>€1,003,200,000</td>
</tr>
<tr>
<td>5</td>
<td>106.3%</td>
<td>84.2%</td>
<td>€700,200,000</td>
<td>€1,255,200,000</td>
</tr>
<tr>
<td>6</td>
<td>94.4%</td>
<td>115.2%</td>
<td>€839,800,000</td>
<td>€1,504,800,000</td>
</tr>
<tr>
<td>7</td>
<td>110.3%</td>
<td>90.1%</td>
<td>€981,500,000</td>
<td>€1,756,650,000</td>
</tr>
<tr>
<td>8</td>
<td>96.1%</td>
<td>60.8%</td>
<td>€1,121,400,000</td>
<td>€2,258,400,000</td>
</tr>
<tr>
<td>9</td>
<td>110.9%</td>
<td>52.6%</td>
<td>€1,160,900,000</td>
<td>€2,008,900,000</td>
</tr>
<tr>
<td>10</td>
<td>90%</td>
<td>-40.9%</td>
<td>€1,403,000,000</td>
<td>€2,510,500,000</td>
</tr>
</tbody>
</table>

Table 6.2: Return from using storage in percentage of investment using price profile 1.

<table>
<thead>
<tr>
<th>Sizing profile #</th>
<th>Return storage 15% ch</th>
<th>Return storage 30% ch</th>
<th>Cost storage 15% ch</th>
<th>Cost storage 30% ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>143.2%</td>
<td>-16.3%</td>
<td>€293,750,000</td>
<td>€513,750,000</td>
</tr>
<tr>
<td>2</td>
<td>126.5%</td>
<td>82.1%</td>
<td>€587,750,000</td>
<td>€1,032,750,000</td>
</tr>
<tr>
<td>3</td>
<td>83.8%</td>
<td>81.9%</td>
<td>€881,500,000</td>
<td>€1,546,500,000</td>
</tr>
<tr>
<td>4</td>
<td>64.4%</td>
<td>53.3%</td>
<td>€118,050,000</td>
<td>€2,065,500,000</td>
</tr>
<tr>
<td>5</td>
<td>79%</td>
<td>40.8%</td>
<td>€147,425,000</td>
<td>€2,584,250,000</td>
</tr>
<tr>
<td>6</td>
<td>44.8%</td>
<td>55.9%</td>
<td>€176,825,000</td>
<td>€3,098,250,000</td>
</tr>
<tr>
<td>7</td>
<td>52.3%</td>
<td>43.7%</td>
<td>€206,700,000</td>
<td>€3,617,000,000</td>
</tr>
<tr>
<td>8</td>
<td>45.6%</td>
<td>29.5%</td>
<td>€236,100,000</td>
<td>€4,136,000,000</td>
</tr>
<tr>
<td>9</td>
<td>52.6%</td>
<td>25.5%</td>
<td>€265,475,000</td>
<td>€4,649,750,000</td>
</tr>
<tr>
<td>10</td>
<td>43.0%</td>
<td>19.8%</td>
<td>€305,375,000</td>
<td>€5,168,750,000</td>
</tr>
</tbody>
</table>

Table 6.3: Return from using storage in percentage of investment using price profile 2.
Table 6.4: Return from using storage in percentage of investment using price profile 3.

<table>
<thead>
<tr>
<th>Sizing profile #</th>
<th>Return storage 15% ch</th>
<th>Return storage 30% ch</th>
<th>Cost storage 15% ch</th>
<th>Cost storage 30% ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.1%</td>
<td>-11.04%</td>
<td>€477,500,000</td>
<td>€807,500,000</td>
</tr>
<tr>
<td>2</td>
<td>77.8%</td>
<td>52.2%</td>
<td>€955,500,000</td>
<td>€1,623,000,000</td>
</tr>
<tr>
<td>3</td>
<td>51.5%</td>
<td>32.1%</td>
<td>€1,433,000,000</td>
<td>€2,430,000,000</td>
</tr>
<tr>
<td>4</td>
<td>37.8%</td>
<td>33.9%</td>
<td>€1,918,500,000</td>
<td>€3,246,000,000</td>
</tr>
<tr>
<td>5</td>
<td>48.6%</td>
<td>26%</td>
<td>€2,396,500,000</td>
<td>€4,061,000,000</td>
</tr>
<tr>
<td>6</td>
<td>27.5%</td>
<td>35.6%</td>
<td>€2,874,000,000</td>
<td>€4,869,000,000</td>
</tr>
<tr>
<td>7</td>
<td>32.2%</td>
<td>27.8%</td>
<td>€3,359,000,000</td>
<td>€5,684,000,000</td>
</tr>
<tr>
<td>8</td>
<td>28%</td>
<td>18.8%</td>
<td>€3,837,000,000</td>
<td>€6,499,000,000</td>
</tr>
<tr>
<td>9</td>
<td>32.4%</td>
<td>16.2%</td>
<td>€314,500,000</td>
<td>€7,307,000,000</td>
</tr>
<tr>
<td>10</td>
<td>26.4%</td>
<td>12.6%</td>
<td>€4,800,000,000</td>
<td>€8,122,500,000</td>
</tr>
</tbody>
</table>

Figure 6.4a and fig. 6.4b show that there is a overall trend towards a lower EENS as the Maximum State of Charge increases. Having said when we look a specific profiles we sometimes see an increase in EENS despite having a higher Maximum State of Charge. This increase can be attributed to the Monte Carlo process that is used and the fact that the subsequent difference in the Maximum State of Charge between profiles are not large enough to entirely overcome the different results following the Monte Carlo process. When looking at the overall result an increase in storage size does clearly lead to reduction in EENS.

The cost of the ESSs that were used are based on the work of [17]. They present 3 different pricing profiles for the storage:

1. 20$/kWh and 500$/kW per unit of storage
2. 50$/kWh and 1000$/kW per unit of storage
3. 100$/kWh and 1500$/kW per unit of storage

The results presented in table 6.1, fig. 6.4a and fig. 6.4b are based on pricing profile 3, the most expensive pricing profile. Since we are only interested in the transmission grid from the perspective of the TSO the ESSs do not necessarily have to turn a profit in order to justify their use by the TSO since their main purposes is to balance the grid. Looking at the different pricing profiles shown in fig. 6.2 and fig. 6.3 we see that price profile 1, 15% ch performs best as the savings across the range of different sizing profiles comes closest to the investment cost of the storage. For this work we have assumed the most expensive price profile. The maximum SoC of the storage was therefore set to 295MWh, the (dis)charge rate was set to 44MW.
Figure 6.2: Sizing of storage with price profile 1
Figure 6.3: Storage sizing with price profile 2
Figure 6.4: Sizing of storage with price profile 3

(a) Investment cost of storage with 15% (dis)charge capability

(b) Investment cost of storage with 30% (dis)charge capability
6.2 Storage siting

In this section we report the results that were achieved by running the simulation using different topologies and storage siting policies. In section 6.2.1 we establish a base line against which to compare the performance of the different topologies. The performances of the siting policies are reported in section 6.2.2. In order to determine the siting of the storage and establish the performance of the topologies and ESSs, we ran the simulation program using a constrained grid, that is to say that we used the capacities and reactance of the lines of the modified IEEE-96 bus used by [12]. It should be noted that the capacity of the lines was multiplied by 2.5. This was required in order to ensure that a solution was feasible for the linear program. This multiplication was needed because the small-world and preferential attachment topologies are not optimal from a power grid point of view due to the random nature in which edges and nodes are assigned. The multiplication was applied to all lines across all topologies in order to stay consistent.

6.2.1 Baseline performance

To establish a baseline against which we can compare the influence of grid topology and the siting policies for ESSs we created two additional topologies with specific properties. These topologies are shown in fig. 3.1 and fig. 3.2 and were created using the process explained in section 4.3.1. In addition to establishing the baseline we also used the results of these simulations to place storages as explained in section 6.2.2. The EENS cost of these simulations are reported in table 6.5.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Cost EENS without storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified IEEE 96</td>
<td>€3,707,181 127</td>
</tr>
<tr>
<td>Small world</td>
<td>€4,094,978 476</td>
</tr>
<tr>
<td>Preferential attachment</td>
<td>€4,582,228 644</td>
</tr>
</tbody>
</table>

Table 6.5: Annual cost of EENS using constrained grids without storage.

6.2.2 Siting policies

The siting of the ESSs is decided in the following manner; having run the simulation we compute the average power flow on every line. We then identify 10 lines which have the highest average power flow on them with respect to their maximum capacity. Based on these 10 lines we then applied the following siting policies:

1. Placing distributed storage units at consumers that contribute the most to the flow on the 10 most used lines. The consumers at which the storage is placed are decided by looking at all consumers nodes in the PTDF and determining which consumer impacts the flow on the corresponding line the most. The reasoning behind this policy is that by placing storage close to consumers we ease their access the electricity which in turn should reduce the EENS.

56
2. Placing distributed storage units at nodes that reduce the flow on a line the most. The nodes at which the storage is placed is decided by looking at all nodes in the PTDF and determining which nodes reduce the flow on the corresponding line the most. The reasoning behind this policy is to see if we can reduce the flow on congested lines.

3. Placing a central storage unit at the node with the highest betweenness centrality. The betweenness centrality metric was chosen for this policy because it allows us to find the node through which most shortest path in the grid pass. Placing an ESS at that location should allow for easy absorption or distribution of power on the grid.

Table 6.6, table 6.7 and table 6.8 show, for each topology the top 10 most used lines and which nodes contributed the most to power flow on these lines.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing consumer node</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 180</td>
<td>29.86</td>
<td>45</td>
</tr>
<tr>
<td>113 170</td>
<td>37.57</td>
<td>56</td>
</tr>
<tr>
<td>118 122</td>
<td>29.39</td>
<td>60</td>
</tr>
<tr>
<td>123 124</td>
<td>28.95</td>
<td>65</td>
</tr>
<tr>
<td>130 132</td>
<td>30.04</td>
<td>70</td>
</tr>
<tr>
<td>132 133</td>
<td>32.95</td>
<td>72</td>
</tr>
<tr>
<td>146 150</td>
<td>29.36</td>
<td>81</td>
</tr>
<tr>
<td>152 173</td>
<td>30.05</td>
<td>85</td>
</tr>
<tr>
<td>163 165</td>
<td>30.46</td>
<td>86</td>
</tr>
<tr>
<td>165 170</td>
<td>44.03</td>
<td>92</td>
</tr>
</tbody>
</table>

Table 6.6: 10 most utilized lines of the modified IEEE 96 topology without storage.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing consumer node</th>
</tr>
</thead>
<tbody>
<tr>
<td>98 99</td>
<td>47.85</td>
<td>84</td>
</tr>
<tr>
<td>103 172</td>
<td>45.08</td>
<td>84</td>
</tr>
<tr>
<td>119 151</td>
<td>81.85</td>
<td>54</td>
</tr>
<tr>
<td>126 127</td>
<td>44.21</td>
<td>84</td>
</tr>
<tr>
<td>134 107</td>
<td>40.84</td>
<td>50</td>
</tr>
<tr>
<td>168 131</td>
<td>48.67</td>
<td>52, 53, 74 and 89</td>
</tr>
<tr>
<td>174 113</td>
<td>44.80</td>
<td>48, 66</td>
</tr>
<tr>
<td>178 179</td>
<td>71.36</td>
<td>67</td>
</tr>
<tr>
<td>179 144</td>
<td>43.99</td>
<td>62</td>
</tr>
<tr>
<td>180 93</td>
<td>65.61</td>
<td>59 and 61</td>
</tr>
</tbody>
</table>

Table 6.7: 10 most utilized lines of the small world topology without storage.
Baseline line utilization

In order to gain additional insight into the impact the topologies and different siting policies we computed histograms for each topology and siting policy. For the histograms we counted how often certain line utilization occurs, spread over 80 bins. The usage count on the vertical axis in the histogram figures indicate how often certain line utilization was counted. The lines that were examined in this manner are the lines that were identified as the most utilized lines in section 6.1. The dotted line in the histogram images indicates the capacity of the line without the 2.5 multiplication that was required in order to run the simulation as explained in section 6.2.

In fig. 6.5 we see the line utilization of the modified IEEE-96 bus. We note that for this topology the selected lines do not reach their actual capacity limit but they do exceed the original capacity limit.

The line utilization of the small world topology, shown in fig. 6.6 seems to be in worse state than that of the modified IEEE-96 bus. Line 119-151 and line 178-179 are most worrying because their utilization occurs mostly on the right side. In addition like the modified IEEE-96 bus all lines exceed their original capacity.

Shown Figure 6.7 is the line utilization of the preferential attachment without storage. It seems to be similar to that of the small world topology shown in fig. 6.6. The full capacity of line 144-105 is used in the vast majority of the simulations, also note the interesting split seen in the exploitation of line 171-94. In addition to this we again see that all lines exceed their original capacity.

Table 6.8: 10 most utilized lines of the preferential attachment topology without storage.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Least contributing node</th>
</tr>
</thead>
<tbody>
<tr>
<td>113 94</td>
<td>34.51</td>
<td>43 and 48</td>
</tr>
<tr>
<td>115 100</td>
<td>36.91</td>
<td>43</td>
</tr>
<tr>
<td>119 115</td>
<td>39.65</td>
<td>70</td>
</tr>
<tr>
<td>134 180</td>
<td>45.29</td>
<td>73</td>
</tr>
<tr>
<td>144 105</td>
<td>82.74</td>
<td>56, 84, 88</td>
</tr>
<tr>
<td>157 150</td>
<td>53.55</td>
<td>68</td>
</tr>
<tr>
<td>162 112</td>
<td>49.80</td>
<td>65</td>
</tr>
<tr>
<td>169 95</td>
<td>45.39</td>
<td>84</td>
</tr>
<tr>
<td>171 94</td>
<td>42.36</td>
<td>58</td>
</tr>
<tr>
<td>180 123</td>
<td>33.19</td>
<td>63 and 73</td>
</tr>
</tbody>
</table>
Figure 6.5: Line exploitation of the modified IEEE topology without storage.

Maximum usage count: 4344
Figure 6.6: Line exploitation of the small world topology without storage.

Maximum usage count: 7128
Figure 6.7: Line exploitation of the preferential attachment topology without storage. Maximum usage count: 4176
Distributed storage near consumer nodes

Table 6.10, table 6.11 and table 6.12 show the line usage of the different topologies where the storage has been placed using siting policy 1. In table 6.9 we see the cost of $EENS$ for each topology and their increase or reduction with respect to the cost of $EENS$ shown in table 6.5. When comparing the lines of each topology with their baseline counterparts we do not see significant changes other than line 180-123 of preferential attachment topology. The average flow on this line increases from 33.190% to 43.356%.

<table>
<thead>
<tr>
<th>Topology</th>
<th>$EENS$ with distributed storage near loads, sizing profile 1, 15% ch</th>
<th>Change of $EENS$ from baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified IEEE 96</td>
<td>€3 303 406 776</td>
<td>10.89% decrease</td>
</tr>
<tr>
<td>Small world</td>
<td>€3 872 696 380</td>
<td>5.42% decrease</td>
</tr>
<tr>
<td>Preferential attachment</td>
<td>€3 830 657 680</td>
<td>16.40% decrease</td>
</tr>
</tbody>
</table>

Table 6.9: Annual cost of EENS using constrained grids with distributed storage near consumers. The baseline we refer to can be found in table 6.5.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing consumer node</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 180</td>
<td>28.72</td>
<td>45</td>
</tr>
<tr>
<td>113 170</td>
<td>37.75</td>
<td>56</td>
</tr>
<tr>
<td>118 122</td>
<td>28.68</td>
<td>60</td>
</tr>
<tr>
<td>123 124</td>
<td>30.17</td>
<td>65</td>
</tr>
<tr>
<td>130 132</td>
<td>29.89</td>
<td>70</td>
</tr>
<tr>
<td>132 133</td>
<td>33.10</td>
<td>72</td>
</tr>
<tr>
<td>146 150</td>
<td>29.48</td>
<td>81</td>
</tr>
<tr>
<td>152 173</td>
<td>30.46</td>
<td>85</td>
</tr>
<tr>
<td>163 165</td>
<td>30.54</td>
<td>86</td>
</tr>
<tr>
<td>165 170</td>
<td>44.14</td>
<td>92</td>
</tr>
</tbody>
</table>

Table 6.10: 10 most used lines of the modified IEEE 96 topology with distributed storage near consumers.
<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing consumer node</th>
</tr>
</thead>
<tbody>
<tr>
<td>98 99</td>
<td>51.46</td>
<td>84</td>
</tr>
<tr>
<td>103 172</td>
<td>48.26</td>
<td>84</td>
</tr>
<tr>
<td>119 151</td>
<td>84.14</td>
<td>54</td>
</tr>
<tr>
<td>126 127</td>
<td>46.16</td>
<td>84</td>
</tr>
<tr>
<td>134 107</td>
<td>43.71</td>
<td>50</td>
</tr>
<tr>
<td>168 131</td>
<td>51.91</td>
<td>52, 53, 74 and 89</td>
</tr>
<tr>
<td>174 113</td>
<td>46</td>
<td>48, 66</td>
</tr>
<tr>
<td>178 179</td>
<td>77.82</td>
<td>67</td>
</tr>
<tr>
<td>179 144</td>
<td>48</td>
<td>62</td>
</tr>
<tr>
<td>180 93</td>
<td>68.12</td>
<td>59 and 61</td>
</tr>
</tbody>
</table>

Table 6.11: 10 most used lines of the small world topology with distributed storage near consumers.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Least contributing node</th>
</tr>
</thead>
<tbody>
<tr>
<td>113 94</td>
<td>35.68</td>
<td>45 and 48</td>
</tr>
<tr>
<td>115 100</td>
<td>36.58</td>
<td>43</td>
</tr>
<tr>
<td>119 115</td>
<td>39.50</td>
<td>70</td>
</tr>
<tr>
<td>134 180</td>
<td>46.12</td>
<td>73</td>
</tr>
<tr>
<td>144 105</td>
<td>84.26</td>
<td>56, 84, 88</td>
</tr>
<tr>
<td>157 150</td>
<td>33.84</td>
<td>68</td>
</tr>
<tr>
<td>162 112</td>
<td>55.17</td>
<td>65</td>
</tr>
<tr>
<td>169 95</td>
<td>50.08</td>
<td>84</td>
</tr>
<tr>
<td>171 94</td>
<td>46.89</td>
<td>58</td>
</tr>
<tr>
<td>180 123</td>
<td>43.35</td>
<td>63 and 73</td>
</tr>
</tbody>
</table>

Table 6.12: 10 most used lines of the preferential attachment topology with distributed storage near consumers.

In fig. 6.8 we see the line utilization of the modified IEEE-96 bus with storage near consumer nodes as identified in table 6.6. Overall there is little change in the way the lines are utilized. For lines 96-180, 118-122, 163-165, we see that the usage count is higher however the overall shape of the distribution on the lines has changed very little. Therefore the higher usage count simply means that the simulation took longer to converge, thus increasing the maximum usage count. Lines 113-170, 130-132, 132-133, 146-150, 165-170 show little difference between having no storage or storage near consumers.

Comparing the line utilization of the small world topology with siting policy 1, shown in fig. 6.9 with the same topology without storage shown in fig. 6.6 we find that there is very little change in the shape of the distributions.

In fig. 6.10 we see the line utilization of the preferential attachment topology with storage near consumers as shown in table 6.8. Comparing this figure with fig. 6.7 we again see very little change in the shape of the distributions.
Figure 6.8: Line utilization of the modified IEEE-96 topology with storage near consumers. Maximum usage count: 4440
Figure 6.9: Line utilization of the small world topology with storage near consumers. Maximum usage count: 4248
Figure 6.10: Line utilization of the preferential attachment topology with storage near consumers. Maximum usage count: 5520
Distributed storage near lowest PTDF node

Corresponding to siting policy 2, table 6.14, table 6.15, and table 6.16 show the flow on the lines previously identified in section 6.2.2. In table 6.13 we see the cost of EENS for the topologies, using siting policy 2, and the decrease or increases of EENS are reported in table 6.5. When comparing average power flow of each topology to their baseline performance we see little change. Line 146-150 in table 6.14 is notable due to its significant decrease when compared to the baseline average power flow of 29.364%.

<table>
<thead>
<tr>
<th>Topology</th>
<th>EENS with distributed storage near loads, sizing profile 1, 15% ch</th>
<th>Change of EENS from baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified IEEE 96</td>
<td>€3 655 185 868</td>
<td>1.40% decrease</td>
</tr>
<tr>
<td>Small world</td>
<td>€4 053 289 745</td>
<td>1.01% decrease</td>
</tr>
<tr>
<td>Preferential attachment</td>
<td>€3 964 089 527</td>
<td>13.48% decrease</td>
</tr>
</tbody>
</table>

Table 6.13: Annual cost of EENS using constrained grids with distributed storage node. The baseline we refer to can be found in table 6.5

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing node</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 180</td>
<td>29.54</td>
<td>180, 195</td>
</tr>
<tr>
<td>113 170</td>
<td>37.18</td>
<td>185</td>
</tr>
<tr>
<td>118 122</td>
<td>29.59</td>
<td>64, 210</td>
</tr>
<tr>
<td>123 124</td>
<td>30.63</td>
<td>68, 206</td>
</tr>
<tr>
<td>130 132</td>
<td>30.90</td>
<td>25</td>
</tr>
<tr>
<td>132 133</td>
<td>33.71</td>
<td>210</td>
</tr>
<tr>
<td>146 150</td>
<td>8.33</td>
<td>32, 82, 83</td>
</tr>
<tr>
<td>152 173</td>
<td>29.95</td>
<td>188, 210</td>
</tr>
<tr>
<td>163 165</td>
<td>30</td>
<td>165</td>
</tr>
<tr>
<td>165 170</td>
<td>43.49</td>
<td>185</td>
</tr>
</tbody>
</table>

Table 6.14: 10 most used lines of the modified IEEE 96 topology with storage near lowest PTDF node.
<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Maximum contributing node</th>
</tr>
</thead>
<tbody>
<tr>
<td>98 99</td>
<td>51.05</td>
<td>36, 72, 183</td>
</tr>
<tr>
<td>103 172</td>
<td>47.27</td>
<td>50, 190</td>
</tr>
<tr>
<td>119 151</td>
<td>82.70</td>
<td>62</td>
</tr>
<tr>
<td>126 127</td>
<td>43.97</td>
<td>127</td>
</tr>
<tr>
<td>134 107</td>
<td>42.75</td>
<td>69</td>
</tr>
<tr>
<td>168 131</td>
<td>50.77</td>
<td>57</td>
</tr>
<tr>
<td>174 113</td>
<td>43.70</td>
<td>58, 76</td>
</tr>
<tr>
<td>178 179</td>
<td>76.41</td>
<td>179</td>
</tr>
<tr>
<td>179 144</td>
<td>47.19</td>
<td>1, 31</td>
</tr>
<tr>
<td>180 93</td>
<td>67.15</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 6.15: 10 most used lines of the small world topology with storage near lowest PTDF node.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
<th>Least contributing node</th>
</tr>
</thead>
<tbody>
<tr>
<td>113 94</td>
<td>34.49</td>
<td>91, 82, 56</td>
</tr>
<tr>
<td>115 100</td>
<td>36.86</td>
<td>29, 69, 86, 186</td>
</tr>
<tr>
<td>119 115</td>
<td>10.01</td>
<td>168</td>
</tr>
<tr>
<td>134 180</td>
<td>46.02</td>
<td>26, 32, 180</td>
</tr>
<tr>
<td>144 105</td>
<td>83.24</td>
<td>11</td>
</tr>
<tr>
<td>157 150</td>
<td>54.55</td>
<td>20, 205</td>
</tr>
<tr>
<td>162 112</td>
<td>49.96</td>
<td>63, 199</td>
</tr>
<tr>
<td>169 95</td>
<td>45.95</td>
<td>33, 89, 54, 207, 187, 184, 183</td>
</tr>
<tr>
<td>171 94</td>
<td>43.02</td>
<td>145</td>
</tr>
<tr>
<td>180 123</td>
<td>33.09</td>
<td>194</td>
</tr>
</tbody>
</table>

Table 6.16: 10 most used lines of the preferential attachment topology with storage near lowest PTDF node.

As seen in fig. 6.11 we see little change on most of the lines when comparing with the modified IEEE-96 bus without storage in terms of the shape of the distributions. Notable exception however is line 146-150 where we that the entire shape of the distribution changes indicating a massive reduction in line utilization.

In fig. 6.12 we see the line utilization of the small world topology with storage placed near nodes with the lowest PTDF values and note that again the overall shape of the distribution has not changed much.

Shown in fig. 6.13 is the line utilization of preferential attachment topology with storage near the node with the lowest PTDF value. The topology shows similar results as the modified IEEE-96 bus. We see that the distribution for line 119-115 has changed significantly meaning that its line utilization is much lower.
Figure 6.11: Line utilization of the modified IEEE-96 topology with storage near lowest PTDF node. Maximum usage count: 6600.
Figure 6.12: Line utilization of the small world topology with storage near lowest PTDF node. Maximum usage count: 3456
Figure 6.13: Line utilization of the preferential attachment topology with storage lowest PTDF node. Maximum usage count: 6960
Centralized storage at highest node centrality

In Table 6.17, we see the cost of EENS and the change in EENS with respect to topologies without storage. We placed the centralized storage at the following inner nodes:

- Modified IEEE-96 bus: Inner node 158
- Small world topology: Inner node 151
- Preferential attachment topology: Inner node 93

<table>
<thead>
<tr>
<th>Topology</th>
<th>EENS with centralized storage at node with highest centrality betweenness, sizing profile 1, 15% ch</th>
<th>Change of EENS from baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified IEEE 96</td>
<td>€3,931,224,203</td>
<td>6.04% increase</td>
</tr>
<tr>
<td>Small world</td>
<td>€4,883,313,167</td>
<td>19.49% increase</td>
</tr>
<tr>
<td>Preferential attachment</td>
<td>€3,917,703,845</td>
<td>14.50% decrease</td>
</tr>
</tbody>
</table>

Table 6.17: Annual cost of EENS using constrained grids with centralized storage at the node with the highest centrality betweenness. The baseline we refer to can be found in Table 6.5.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 180</td>
<td>28.16</td>
</tr>
<tr>
<td>113 170</td>
<td>36.37</td>
</tr>
<tr>
<td>118 122</td>
<td>28.58</td>
</tr>
<tr>
<td>123 124</td>
<td>29.48</td>
</tr>
<tr>
<td>130 132</td>
<td>29.60</td>
</tr>
<tr>
<td>132 133</td>
<td>32.64</td>
</tr>
<tr>
<td>146 150</td>
<td>29.13</td>
</tr>
<tr>
<td>152 173</td>
<td>29.74</td>
</tr>
<tr>
<td>163 165</td>
<td>29.66</td>
</tr>
<tr>
<td>165 170</td>
<td>42.86</td>
</tr>
</tbody>
</table>

Table 6.18: 10 most used lines of the modified IEEE 96 topology with a storage node placed at the node with highest centrality betweenness.
Table 6.19: 10 most used lines of the small world topology with a storage node placed at the node with highest centrality betweenness

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>98 99</td>
<td>48.42</td>
</tr>
<tr>
<td>103 172</td>
<td>45.98</td>
</tr>
<tr>
<td>119 151</td>
<td>81.77</td>
</tr>
<tr>
<td>126 127</td>
<td>44.65</td>
</tr>
<tr>
<td>134 107</td>
<td>41.30</td>
</tr>
<tr>
<td>168 131</td>
<td>50.01</td>
</tr>
<tr>
<td>174 113</td>
<td>45.29</td>
</tr>
<tr>
<td>178 179</td>
<td>72.96</td>
</tr>
<tr>
<td>179 144</td>
<td>45.01</td>
</tr>
<tr>
<td>180 93</td>
<td>67.13</td>
</tr>
</tbody>
</table>

Table 6.20: 10 most used lines of the preferential attachment topology with a storage node placed at the node with highest centrality betweenness.

<table>
<thead>
<tr>
<th>Line from node i to j</th>
<th>Yearly average power flow as % of line capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>113 94</td>
<td>35.11</td>
</tr>
<tr>
<td>115 100</td>
<td>36.84</td>
</tr>
<tr>
<td>119 115</td>
<td>39.93</td>
</tr>
<tr>
<td>134 180</td>
<td>46.25</td>
</tr>
<tr>
<td>144 105</td>
<td>84.25</td>
</tr>
<tr>
<td>157 150</td>
<td>54.66</td>
</tr>
<tr>
<td>162 112</td>
<td>49.50</td>
</tr>
<tr>
<td>169 95</td>
<td>46.88</td>
</tr>
<tr>
<td>171 94</td>
<td>42.32</td>
</tr>
<tr>
<td>180 123</td>
<td>32.81</td>
</tr>
</tbody>
</table>

As shown in fig. 6.14 the introduction of a central storage node does little to change the shape of the distributions.

In fig. 6.15 we see the line utilization for the small world topology with central storage. Compared to fig. 6.6 we see that the shape of the distribution has not changed.

Shown in fig. 6.16 is the line utilization of the preferential attachment topology with centralized storage. The centralized storage has a similar affect, in terms of line utilization on this topology as it had on the small world one. The overall shape of the distributions does not change.
Figure 6.14: Line utilization of the modified IEEE-96 topology with storage near the node with highest centrality betweenness. Maximum usage count: 5472.
Figure 6.15: Line utilization of the small world topology with storage near the node with highest centrality betweenness. Maximum usage count: 3744
Figure 6.16: Line utilization of the preferential topology with storage near least contributing node. Maximum usage count: 7272
6.3 Graph metrics of the topologies

As explained in chapter 2, we have defined several metrics that allow us to differentiate between different topologies. For each of the topologies these metrics are:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average degree</td>
<td>2.321</td>
</tr>
<tr>
<td>Characteristic path length</td>
<td>0.0385</td>
</tr>
<tr>
<td>Average clustering</td>
<td>0.0</td>
</tr>
<tr>
<td>Small world property</td>
<td>0.0</td>
</tr>
<tr>
<td>Diameter</td>
<td>17.0</td>
</tr>
</tbody>
</table>

Table 6.21: Metrics of the modified IEEE-96 bus shown in fig. 6.1

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average degree</td>
<td>2.330</td>
</tr>
<tr>
<td>Characteristic path length</td>
<td>0.034</td>
</tr>
<tr>
<td>Average clustering</td>
<td>0.030</td>
</tr>
<tr>
<td>Small world property</td>
<td>2.869</td>
</tr>
<tr>
<td>Diameter</td>
<td>16.0</td>
</tr>
</tbody>
</table>

Table 6.22: Metrics of the small world topology shown in fig. 3.1

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average degree</td>
<td>2.330</td>
</tr>
<tr>
<td>Characteristic path length</td>
<td>0.0287</td>
</tr>
<tr>
<td>Average clustering</td>
<td>0.0173</td>
</tr>
<tr>
<td>Small world property</td>
<td>1.816</td>
</tr>
<tr>
<td>Diameter</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 6.23: Metrics of the preferential attachment topology shown in fig. 3.2

We note that both the small world topology and the preferential attachment topology are said to be small world topologies because their small world property is greater than 1. Having said that because the small world topology has a larger small world property than the preferential attachment topology we consider it to be more small world than the latter. We do not consider the modified IEEE-96 bus to be small world, this is due to the absence of clustering, causing the small world property to be 0. The average degree across all of the topologies is approximately the same. The modified IEEE-96 bus and the small world topology have a somewhat similar characteristic path length and the preferential attachment topology is least similar.
Chapter 7

Conclusion

In the introduction of this work we formulated the following research questions:

- What is the optimal size of the storage to minimize EENS and storage cost?
- How does the topology of the network influence the operation of storage?

We have aimed to answer these questions by further developing the work of [12] by introducing Monte Carlo techniques as specified in chapter 3, chapter 4, and chapter 5. By introducing Monte Carlo techniques into the simulation we have effectively simulated different scenarios in which the production of generators and consumption of consumers changes. Using the simulation program we have shown that the sizing of the storage depends on its investment cost and the cost of EENS. In this work we assumed the most expensive price profile for the storage and we found that when considering the saving in EENS a storage with a capacity of 295MWh and a (dis)charge rate of 15% of its Maximum State of Charge is feasible.

Using storage profile 1 15% ch from table 6.1 we applied three different storage siting policies and examined their effectiveness. As shown in section 6.2 siting policy 1 is most effective in reducing EENS followed by siting policy 2. Siting policy 3 performs worst, increasing the EENS on the modified IEEE-96 bus and small world topology and decreasing the EENS on the preferential attachment topology. When examining the average flow we found little change. Exceptions however are line 146-150 on the modified IEEE-96 bus with siting policy 2 applied and line 119-115 on the preferential attachment topology again with siting policy 2 applied. For the aforementioned lines we see a significant reduction in line utilization as shown in fig. 6.11 and fig. 6.13. Other siting policies and topologies do not show such a decrease in line utilization.

When we examine each of the topologies without storage we find that the modified IEEE-96 bus has the lowest EENS followed by the small world topology and the preferential attachment topology as shown in table 6.3. The introduction of storage has the biggest impact on the preferential attachment topology regardless of siting policy in terms of EENS reduction. When we consider the values of EENS as opposed to the reductions the picture is different in that the modified
IEEE-96 bus has the lowest $\text{EENS}$ followed by the small world topology and finally the preferential attachment topology. When we introduce storage using siting policies 1 or 2 from section 6.2.2, we find that the modified IEEE-96 bus still has the lowest $\text{EENS}$ followed by the preferential attachment topology and then the small world topology. When we introduce storage according to siting policy 3, we find that for the small world topology and the modified IEEE-96 bus $\text{EENS}$ increases. Only the preferential attachment topology shows a decrease in $\text{EENS}$ when siting policy 3 is applied.
Chapter 8

Discussion

In this chapter we discuss the results and conclusions as reported in Chapter 6 and Chapter 7. We begin by discussing the results of the sizing in section 8.1 followed by a discussion of the siting policies in section 8.2. We finish this chapter with a discussion about the influence of grid topology on EENS in section 8.3.

8.1 Sizing of storage

When examining the results of the sizing we found that having storage with a 15% (dis)charge rate results in better performance than storage with a 30% (dis)charge rate. One might think that a larger (dis)charge rate automatically improves the performance of storage by making it more versatile. Our simulation however shows the opposite; a storage with a slow (dis)charge performs better than a fast (dis)charge rate. The reason behind this is that the cost of having a higher (dis)charge rate cannot be offset by savings in EENS. This lack of additional savings can be explained in the following manner; if the capacity of our storage is 295 MWh and our (dis)charge rate is 88 MW then fully using the discharge rate will drain the storage in 3.35 hours. If our discharge rate is 44 MW it will take 6.70 hours to fully drain the storage. We speculate that because EENS cannot be compensated after the 3.35 hours of discharging the cost of subsequent EENS is too high to compensate the higher investment cost of fast (dis)charging. The opposite is said to be true of slow (dis)charging, in which case we are able to minimize EENS over longer periods of time while having lower investment cost. It should be noted that charging the storage during the day time may be unfeasible for longer periods of time due to the fact that supplying consumers with power takes priority over charging the storage. Therefore situations may arise in which an initial discharging of the storage will result in the storage remaining drained for the remainder of the day. In such situations having a slow discharge rate will allow for better utilization of the storage.
8.2 The influence of topology on storage performance

In this section we discuss the results of introducing storage into the grid for each of the topologies and siting policies. In order to frame this discussion we also discuss what occurs when we run the simulation program without introducing ESS.

8.2.1 Without Storage

To begin our discussion let us first examine the topologies without storage. The modified IEEE-96 bus has the lowest EENS followed by the small world topology and the preferential attachment topology. The difference in performance between the modified IEEE-96 bus, the small world topology and the preferential attachment topology can be explained by the fact that the modified IEEE-96 bus has undergone optimization. This optimization means that the configuration of the edges connecting inner nodes is more optimal than in the small world and preferential attachment topology. On the modified IEEE-96 bus there are three areas of consumers and generators, transferring power between these areas is relatively easy because of the optimization of the edges connecting them. Similar areas exist on the small world topology and the preferential attachment topology has hubs to which many nodes are connected. The reason for the poor performance of the small world and preferential attachment topologies is that these topologies have not undergone any optimization. Therefore the edges connecting areas or hubs may be of insufficient capacity, leading to EENS.

When we examine the difference in performance between the small world and preferential attachment topologies without storage we find that the small world topology performs better. A possible reason for this is that generator and consumer nodes are more evenly distributed within the small world topology than they are in the preferential attachment topology as seen in fig. 3.1 and fig. 3.2. Within the preferential attachment topology we see more lines that can become constrained because they connect an outer node to an inner node, which is then connected to the grid. An example of this is conventional generator 0 and consumer 42 in fig. 3.2. Such connections do not occur on the modified IEEE-96 bus and small world topology. This argument is supported by the average clustering metric and small world property shown in table 6.22 and table 6.23. A high average clustering and small world measurement indicate a more closely connected topology. As a general rule the more connected a topology is the easier it is to transfer power to locations where it is needed.

8.2.2 Siting policy 1: Storage near consumers

For this siting policy we see the largest reduction in EENS on the preferential attachment topology, followed by the modified IEEE-96 bus, the smallest reduction occurs on the small world topology. Overall we see very minor changes in the average power flow on the 10 most utilized lines. The largest difference is seen on the preferential attachment topology on line 180-123 which has an average flow without storage of 33.19% which is increased with storage to 43.35%. Such a raise is expected because placing storage near consumers should increase
the average power flow because the total flow towards consumers increases due to the night charging policy of the storage. We found that this siting policy leads to significant reductions in $EENS$ for all the topologies. This means that siting policy 1 is an effective way to reduce $EENS$. In terms of line utilization we have not seen a significant change in the way the lines are used.

8.2.3 Siting policy 2: Storage near least contributing node

As explained in section 8.2.2 this policy is meant to decrease the power flow on congested lines. Therefore we expect the average power flow to decrease and the distributions to shift to the left of the histograms shown in fig. 6.11, fig. 6.12, and fig. 6.13. First let us note that that we are still reducing $EENS$ however we do so much less efficiently than siting policy 1. Notably the reduction of $EENS$ is much lower for the modified IEEE-96 bus and small world topology. The preferential attachment topology has around the same reduction in $EENS$ as in siting policy 1. When we consider the average power flow results and line utilization we find little change for the majority of the lines. However line 146-150 from the modified IEEE-96 bus shows a large shift to the left of the histogram as does line 119-115 on preferential attachment topology. Moreover the average power flow on these lines has been lowered significantly from respectively from 29.39% to 8.33% and from 39.65% to 10.01%. Therefore we have shown that placing storages according to siting policy 2 does reduce flow on congested lines. The reason that line 146-150 from the modified IEEE-96 bus and line 119-115 from the preferential attachment topology show a significant reduction can be attributed to specifics in the topologies. The PTDF allows us to determine how the injection of power at node A and withdraw of power at node B is going to affect a line. However this does not provide a guarantee that placement of storage using the PTDF is going to cause the desired effect. It is possible that the flow on a line cannot be influenced by placing storage if that line is subject to many influences from other lines or nodes.

8.2.4 Siting policy 3: Centralized storage

This siting policy has the worst performance it increases $EENS$ on the modified IEEE-96 bus and small world topology. Even though we see a decrease in $EENS$ on the preferential attachment topology this decrease is smallest when compared to the other siting policies. When examining the 10 most utilized lines we again find that the introduction of storage has minimal impact on the average power flow. In addition to this the shape of the distributions in the line utilizations shows minimal change.

The fact that $EENS$ increases on the modified IEEE-96 bus and small world topology might be caused by the additional power flow to and from the centralized storage. This causes congestion on critical lines near the node with the highest centrality betweenness. This congestion then forces the power to take an alternate route that is not as efficient. This shift in power flow may therefore be responsible for the increase in $EENS$. The reason we do not see such an effect on the preferential attachment topology is that the characteristic path length is shorter as is the diameter. This means that the power flow to from generators to consumers needs to travel smaller distances, as a result the power flow on the
preferential attachment topology is not influenced by the topology. Both in the modified IEEE-96 bus and small world topology the characteristics path length and the diameter are larger causing the power flow to be more subjected to the grid topology.

8.3 Expected Energy Not Supplied and grid topology

When examining the results of the siting policies in terms on EENS shown in table 6.5 we see that the modified IEEE-96 bus has the lowest EENS followed by the small world topology and the preferential attachment topology. When we then examine the EENS following the introduction of storages shown in table 6.9 table 6.13 and table 6.17 we find that the preferential attachment topology has the largest reductions for each of the siting policies. This raises the questions; why does EENS decrease the most on the preferential attachment topology? And why is its performance without storage is worst? The preferential attachment topology has hubs of consumers and generators and the transfer of power within these hubs is very efficient. However, when a certain hub is under or over producing we have to transfer power to or from another hub. The transfer of power between hubs might be difficult to handle for the preferential attachment topology because the collective load of a hub is much higher than that of an individual consumer or generator as we find in the modified IEEE-96 bus and small world topology. The difficulty in transferring power from hub to hub also explains why we see the largest reduction in EENS on the preferential attachment topology. The storage is charged during the night period during which the load is usually low. The charging of the storage during non-peak times allows us to discharge the storage during peak times. Given that the storages are in most cases connected directly to hubs the (dis)charging of storage effectively allows us to reduce the reliance of the power flow on the topology outside of these hubs.

When we examine the differences between the performance of the modified IEEE-96 bus and the small world topology we find that in terms of EENS the modified IEEE-96 bus outperforms the small world topology in every storage siting policy. In addition when running the simulation without storage the modified IEEE-96 bus also outperforms the small world topology. When we examine the differences between these two topologies we note the absence of the small world property and the absence of clustering on the modified IEEE-96 bus. Both of these metrics are present on the small world topology, clustering however in theory should cause a decrease in EENS because it is an indication of closer connectivity of the topology. Having said that it is important to note that the modified IEEE-96 bus, while synthetic is the most realistic topology and it has undergone some optimization as part of that realism. This optimization is not present on the small world topology, we therefore conclude that the performance difference between these two topologies are due to the optimization of the modified IEEE-96 bus and lack thereof on the small world topology. In addition the clustering within the small world topology may be too local to areas to meaningfully impact the transfer of power between clusters, thus clustering may not affect EENS much.
Chapter 9

Future work

In many cases the conclusion of a research project lead to more questions, different avenues of research and future work. This work is not an exception, therefore we present future works and questions that could be performed and answered within the context of this thesis.

First is the questions of centralized storage; in this work the centralized storage performs worst in terms of $E_{ENS}$. What remains to be seen however if this lackluster performance is due to the centralized storage itself or the siting policy that was used to place the storage. Therefore a future research project could include different siting policies for the centralized storage. Such policies might include placing storage in areas where $E_{ENS}$ and curtailment occurs as opposed to the siting policies that we have used. What also remains to be determined is if the performance difference between the modified IEEE-96 bus and small world topology can indeed be attributed to the optimization that the modified IEEE-96 bus has undergone. It is therefore advisable to do such optimization on the small world topology as well and then determine which topology performs better. In doing so the effects caused by clustering in the small world topology will also become more clear.

In this work we have used $E_{ENS}$ as a metric in order to judge the performance of the different topologies and siting policies. The decrease of $E_{ENS}$ however does not paint the whole picture that is produced by the simulation because the objective function minimized $E_{ENS}$ and curtailment. Therefore in order to improve the quality of we judge the performance of the storages we can include curtailment. This would of course also impact the sizing of storage because when we include curtailment it allows us to take into the savings from reducing $E_{ENS}$ and savings from reducing curtailment. Of course in order to do so one must first establish a realistic economical foundation that determines the savings from reducing curtailment.

While the modified IEEE-96 bus is somewhat realistic the small world topology and preferential topology are both completely synthetic. Therefore one of the most interesting future approaches would be to create a topology that represents a real transmission grid. Using this topology and the simulation program that we have developed we could validate our results and determine if they apply to real life topologies as opposed to only synthetic topologies.
Bibliography


Acronyms

**EENS** Expected Energy Not Supplied. 1 10 11 13 20 29 31 41 42 47 55 57 59 61 72 79 86 93 100 104 Glossary: Expected Energy Not Supplied

**ESS** Energy Storage System. 7 9 11 14 16 18 30 31 33 35 38 41 53 56 64 65 97 106

**MCD** Monte Carlo Draw. 13 32 34 35 51 53 104 Glossary: Monte Carlo Draw

**MPPT** Maximum Power Point Tracker. 13

**MTTF** Mean Time To Failure. 22 51 104 Glossary: Mean Time To Failure

**MTTR** Mean Time To Repair. 22 104 Glossary: Mean Time To Repair

**NGO** Non-Government Organization. 9

**PTDF** Power Transfer Distribution Factor. 7 12 54 61 72 79 82 84 94 98 104 Glossary: Power Transfer Distribution Factor

**PVS** Photovoltaic System. 13 50 51

**SoC** State of Charge. 23 28 34 35 56 60 104 Glossary: State of Charge

**TSO** Transmission System Operator. 32 33 52 64 60 104 Glossary: Transmission System Operator

**WECS** Wind Energy Conversion System. 13
Glossary

**Day ahead limit minimum** Limits the maximum production to 7.5% above the minimum production capacity. 22

**Day ahead production maximum** Limits the maximum production to 7.5% below the maximum production capacity. 22

**Expected Energy Not Supplied** The amount of energy that is not supplied to consumers. 1, 104

**Expected load** The expected load is a forecast of the actual load that will be placed on the grid. Using the expected load we perform day ahead production planning. 32

**Maximum Production** The maximum production capacity of a generator in MW. 22

**Maximum Production change** Limits the production increases of conventional generators due to their spinup time; the maximum increase or decrease is 50% of the maximum production. 22

**Maximum State of Charge** The maximum energy capacity of a storage node. 23, 31, 53, 56, 59

**Mean Time To Failure** The average duration it takes for a generator to fail. 22, 104

**Mean Time To Repair** The duration it takes to repair a generator after failure. 22, 104

**Minimum Production** The minimum production capacity of a generator in MW. 22

**Minimum State of Charge** The minimum State of Charge of a storage node. Discharging beyond this point will reduce the lifespan of the storage node and is therefore not desirable. 23

**Monte Carlo Draw** A random number drawn from a certain distribution (normal, weibull, etc). 13, 104

**Power Transfer Distribution Factor** A matrix that represent the linearized impact of a transfer of power between a producer and consumer. 7, 104
**Real load** The load that is placed on the grid during real time operation. 32

**State of Charge** The amount of energy a ESS holds. 23 [104]

**Transmission System Operator** The Transmission System Operator is responsible for ensuring the operations of the transmission grid. 32 [104]
Appendices
Appendix A

Input

A.1 Hourly load fall

5685,
5362,
5216,
5175,
5209,
5453,
6391,
7437,
8127,
8353,
8471,
8546,
8049,
7968,
8184,
8223,
8218,
7935,
8268,
8559,
8134,
7571,
6830,
6255

A.2 Hourly load spring

5832,
5496,
5322,
5220,
5268,
A.3 Hourly load summer

6926,
6536,
6331,
6194,
6181,
6276,
6741,
7701,
8753,
9350,
9595,
9716,
9360,
9325,
9536,
9562,
9588,
9348,
8963,
8726,
8640,
8586,
8090,
7512

A.4 Hourly load winter
A.5 Configuration file

```{ general : {
   modelday-file: "../modelday.mod",
   modelnight-file: "../modelnight.mod",
   input-file: "../network.0.csv",
   output-folder: "../output/",
   graphstate-folder: "../graphstate/",
   simulation-runs: 1000,
   numberOfTimeSteps : 24,
   dailyMaxLoadDemand : 9716, # according to season
   durationOfEachStep : 1,
   costLoadShedding : 3000,
   costCurtailment : 1500, #original was 200.0F
   efficiency : 75,
   EENSConvergenceThreshold : 100,
   cleanupHourlyOutput: false,
},
  monte-carlo : {
   //For weibull distribution: alpha = 1.6, beta = 8
   shape : 1.6,
   scale : 8,
   //For normal distribution: mean = 0, sigma = 0.05
   mean : 0,
```

94
\[ \sigma: 0.04 \]

**conventionalGenerator:**

- \( \text{maxProductionIncrease} : 0.5 \) // \( |Ph - Ph+1| < 0.5\% \ P_{max} \)
- \( \text{dayAheadLimitMax} : 0.075 \) // \( p_{min} + 7.5\% | p_{max} - 7.5\% \)
- \( \text{dayAheadLimitMin} : 0.075 \) // \( p_{min} + 7.5\% | p_{max} - 7.5\% \)
  
  \( \text{mttf} : 630, \)
  \( \text{mttr} : 60, \)

**disableCheapGeneratorThresholdMaxP:** 60

**load-curves:**

- \( \text{spring} : "\text{Expected Load spring.csv}" \)
- \( \text{summer} : "\text{Expected Load summer.csv}" \)
- \( \text{fall} : "\text{Expected Load fall.csv}" \)
- \( \text{winter} : "\text{Expected Load winter.csv}" \)

**oilOffer:**

- \( \text{percentageFirstIncrease} : 0.4 \)
- \( \text{percentageFirstDecrease} : 0.6 \)
- \( \text{priceIncreaseOne} : 10, \)
- \( \text{priceIncreaseTwo} : 20, \)
- \( \text{priceDecreaseOne} : 10, \)
- \( \text{priceDecreaseTwo} : 20 \)

**coalOffer:**

- \( \text{percentageFirstIncrease} : 0.4 \)
- \( \text{percentageFirstDecrease} : 0.6 \)
- \( \text{priceIncreaseOne} : 15, \)
- \( \text{priceIncreaseTwo} : 25, \)
- \( \text{priceDecreaseOne} : 15, \)
- \( \text{priceDecreaseTwo} : 25 \)

**nuclearOffer:**

- \( \text{percentageFirstIncrease} : 0.4 \)
- \( \text{percentageFirstDecrease} : 0.6 \)
- \( \text{priceIncreaseOne} : 8, \)
- \( \text{priceIncreaseTwo} : 22, \)
- \( \text{priceDecreaseOne} : 26, \)
- \( \text{priceDecreaseTwo} : 30 \)

**hydroOffer:**

- \( \text{percentageFirstIncrease} : 0.4 \)
- \( \text{percentageFirstDecrease} : 0.6 \)
- \( \text{priceIncreaseOne} : 5, \)
- \( \text{priceIncreaseTwo} : 10, \)
- \( \text{priceDecreaseOne} : 5, \)
- \( \text{priceDecreaseTwo} : 10 \)

**hydroelectricGenerator:**

- \( \text{mttf} : 1000, \)
mtr: 60
}
windGenerator :{
  vCutIn: 3,
  vCutOff: 25,
  vRated: 12,
  pRated: 220
},
solarGenerator :{
  panelEfficiency: 0.15,
  irradianceConstant: 1362, // Solar constant
eccentricity: 0.033, // Eccentricity correction
  Factor
  longitude: 53.218705, // Position of solar parks
  longitude: 6.567793,
  year: 2016
},
Storage :{
  chargeEfficiencyOfStorage: 0.87,
  dischargEfficiencyOfStorage: 0.87,
  beginChargeTime: 23,
  endChargeTime: 5
},
glpso-config :{
  outpath1: "input",
  outpath2: "mod",
  solpath1: "glpsol -d", // adjust this to the correct
  version of glpsol <= 4.52
  solpath2: "-m"
}

A.6 Model night

param n_inner;
param n_cons; # Number of consumer nodes
param n_tgen; # number of conventional generators
param n_rgen; # number of renewable generators
param n_storage; # number of storage nodes
param n_tot; # total number of nodes.
param m_factor; # multiplication factor per-unit MW
param pi; #pi constant
param totload; # total demand for consume rnodes
param cost_curt; # renewable cut costs
param cost_sl; # cost shedded load
param outname;

param current_hour;
param start_charge_time;
param end_charge_time;

96
param n;

#initialize variables
set nodes := 0..n_tot;
set tgen := 0..n_tgen;
set consumers := n_tgen+1..(n_tgen+n_cons);
set inner := (n_tgen+n_cons+1)..(n_tgen+n_cons+n_inner);
set rgen := (n_tgen+n_cons+n_inner+1)..(n_tgen+n_cons+n_inner+n_rgen);
set storage := (n_tgen+n_cons+n_inner+n_rgen+1)..n_tot;

param weight {nodes, nodes} >=0;
param capacity {nodes, nodes} >=0;
param costs {tgen} >=0, default 0;
param rcost {rgen} >=0, default 0;
param rprodmax {rgen} >=0;
param rprodmin {rgen} >=0;
param storgemin {storage};
param storgemax {storage};
param minprod {tgen} >=0;
param maxprod {tgen} >=0;
param loads {consumers} >=0;
param production {tgen} >= 0;
param planned_production {tgen} >=0;
param rewproduction {rgen};
param flowfromstorage {storage};

#phase angle constraint
var theta {nodes} >= -pi/2, <= pi/2;

#The function to minimize daily system cost.
minimize obj : 
(\sum\{i in rgen\}
   cost_curt \ast (rprodmax[i] - (\sum\{j in nodes : capacity[i,j] <> 0\} \ast (\theta[i]-\theta[j]) / weight[i,j]) \ast m_factor)) + 
(\sum\{i in consumers\}
   cost_sl \ast (loads[i] - (\sum\{j in nodes : capacity[i,j] <= 0\} \ast (\theta[i]-\theta[j]) / weight[i,j]) \ast m_factor));

#subject to these constrains
subject to anglestability :
   theta[46], = 0;

#Maximum flow rate
subject to flowcapmax { i in nodes, j in nodes : capacity[i,j] <= 0} :
   ((\theta[i]-\theta[j]) / weight[i,j]) \ast m_factor, <=
capacity[i,j];

# Minimum flow rate
subject to flowcapmin { i in nodes, j in nodes : capacity[i,j] <> 0}:
  ((theta[i] - theta[j]) / weight[i,j]) * m_factor, >= -capacity[i,j];

# Flow conservation on an inner node
subject to flowcons { i in inner}:
  sum { j in nodes : capacity[i,j] <> 0} ((theta[i] - theta[j]) / weight[i,j]) * m_factor, =
  sum { j in nodes : capacity[j,i] <> 0} ((theta[j] - theta[i]) / weight[j,i]) * m_factor;

# Renewable production constraint
subject to setRewProduction { i in rgen}:
  sum { j in nodes : capacity[i,j] <> 0} ((theta[i] - theta[j]) / weight[i,j]) * m_factor, <=
  rewproduction[i];

# Renewable production constraint
subject to minRewProduction { i in rgen}:
  sum { j in nodes : capacity[i,j] <> 0} ((theta[i] - theta[j]) / weight[i,j]) * m_factor, >= 0;

# Traditional production constraint
subject to genproduction { i in tgen}:
  sum { j in nodes : capacity[i,j] <> 0} ((theta[i] - theta[j]) / weight[i,j]) * m_factor, =
  production[i];

subject to flowfromstorageNight { i in storage }:
  sum { j in nodes : capacity[i,j] <> 0} ((theta[i] - theta[j]) / weight[i,j]) * m_factor, =
  flowfromstorage[i];

# The amount of energy send to a consumer should be lower or equal to the load of the consumer
subject to loadfix { i in consumers}:
  sum { j in nodes : capacity[j,i] <> 0} ((theta[j] - theta[i]) / weight[j,i]) * m_factor, <= loads[i];

# Flow to a consumer can never be lower than zero
subject to loadMinValue { i in consumers}:
  sum { j in nodes : capacity[j,i] <> 0} ((theta[j] - theta[i]) / weight[j,i]) * m_factor, >= 0;
# Balance supply and demand of energy.

subject to prodlodeq :

\[ \text{sum\{i in (r_gen union t_gen union storage), j in nodes : capacity[i,j] <> 0\}} \]
\[ (\frac{\text{theta}[i]-\text{theta}[j]}{\text{weight}[i,j]} \times \text{m_factor}) = \text{sum\{i in consumers, j in nodes : capacity[j,i] <> 0\}} \]
\[ (\frac{\text{theta}[j]-\text{theta}[i]}{\text{weight}[j,i]} \times \text{m_factor}) \]

# go ahead and solve the model

solve;

# display {i in nodes, j in nodes : capacity[i,j] <> 0}: i, j, (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j] \times \text{m_factor};

printf {i in storage, j in nodes : capacity[i,j] <> 0}:
"\%d \%d %.3f
", i, j, (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j] \times \text{m_factor} > "update.txt";

# For consumer nodes, generator nodes, renewable generator nodes

printf {i in nodes, j in nodes : capacity[i,j] <> 0}:
"\%d \%d %.4f, %.4f, %.4f
", i, j, (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j] \times \text{m_factor}, (\text{abs}((\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j]) \times \text{m_factor} / \text{capacity[i,j]} \times 100, (\text{abs}((\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j]) \times \text{m_factor} / \text{totload}) \times 100 > "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";

# Renewable energy curtailment

printf {i in r_gen}:
"R, \%d, %.4f
", i, (\text{sum\{j in nodes : capacity[i,j] <> 0\}} (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j]) \times \text{m_factor} >> "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";

# Traditional generators

printf {i in t_gen}:
"TG, \%d, %.4f
", i, (\text{sum\{j in nodes : capacity[i,j] <> 0\}} (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j]) \times \text{m_factor} >> "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";

# Consumers

printf {i in nodes, j in consumers : capacity[i,j] <> 0}:
"C, \%d, %.4f
", i, j, (\text{theta}[i] - \text{theta}[j])/\text{weight}[i,j] \times \text{m_factor} >> "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";

# inner nodes
printf {i in inner} : "I,%d,% . 4f \n", i, (sum{j in nodes : capacity[i,j] <> 0} ((theta[i]−theta[j])/ weight[i,j] ) * m_factor) >> "sol" & outname & ".txt";
printf : "\n" >> "sol" & outname & ".txt";

printf {i in storage} : "Stor,%d,% . 4f \n", i, (sum{j in nodes : capacity[i,j] <> 0} ((theta[i]−theta[j])/ weight[i,j] ) * m_factor) >> "sol" & outname & ".txt";
printf : "\n" >> "sol" & outname & ".txt";

printf "CURTAILMENT,% . 4f \n", (sum{i in rgen} (rprodmax[i] -(sum{j in nodes : capacity[i,j] <> 0} (theta[i]−theta[j])/ weight[i,j] ) * m_factor)) >> "sol" & outname & ".txt";

printf "EENS,% . 4f \n", (sum{i in consumers} (loads[i] -(sum{j in nodes : capacity[i,j] <> 0} (theta[i]−theta[j])/ weight[i,j] ) * m_factor)) >> "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";
#endif
printf {i in nodes, j in nodes : capacity[j,i] <> 0} : "
    flow in [%d , %d ] = % .6f \n", j, i, ((theta[j]−theta[i])/ weight[j,i] ) * m_factor;
#endif
printf {i in nodes} : "theta %d = % .6f \n", i, theta[i] >> "sol" & outname & ".txt";
#endif
printf {i in rgen, j in nodes : capacity[i,j] <> 0} : "
    Theta: % .4f , % .4f \n", theta[i], theta[j];
#endif

A.7 Model day

param n_inner;
param n_cons; #Number of consumer nodes
param n_tgen; #number of conventional generators
param n_rgen; #number of renewable generators
param n_storage; #number of storage nodes
param n_tot; # total number of nodes
param m_factor; #multiplication factor per-unit MV
param pi; #pi constant
param totload; #total demand for consume rnodes
param cost_curt; # renewable cut costs
param cost_sl; # cost shedded load
param outname;

param current_hour;
param start_charge_time;
param end_charge_time;
param n;

# initialize variables
set nodes := 0..n_tot;
set tgen := 0..n_tgen;
set consumers := n_tgen+1..(n_tgen+n_cons);
set inner := (n_tgen+n_cons+1)..<(n_tgen+n_cons+n_inner);
set rgen := (n_tgen+n_cons+n_inner+1)..<(n_tgen+n_cons+n_inner+n_rgen);
set storage := (n_tgen+n_cons+n_inner+n_rgen+1)..<n_tot;

param weight {nodes, nodes} >= 0;
param capacity {nodes, nodes} >= 0;
param costs {tgen} >= 0, default 0;
param rcost {rgen} >= 0, default 0;
param rprodmax {rgen} >= 0;
param rprodmin {rgen} >= 0;
param storagemin {storage};
param storamemax {storage};
param mintprod {tgen} >= 0;
param maxtprod {tgen} >= 0;
param loads {consumers} >= 0;
param production {tgen} >= 0;
param planned_production{tgen} >= 0;
param rewproduction {rgen};
param flowmaxcharge {storage};
param flowmaxdischarge {storage};

# phase angle constraint
var theta {nodes} >= -pi/2, <= pi/2;

# The function to minimize daily system cost.
minimize obj :

(sum{i in rgen} cost_curt * (rprodmax[i] - (sum{j in nodes : capacity[i,j] <> 0} (theta[i]-theta[j])/weight[i,j])*m_factor)) +
(sum{i in consumers} cost_sl * (loads[i] - (sum{j in nodes : capacity[i,j] <> 0} (theta[i]-theta[j])/weight[i,j])*m_factor));

# Subject to these constraints
subject to anglestability:
  \( \theta_{[46]} = 0; \)

# Maximum flow rate
subject to flowcapmax \{ i in nodes, j in nodes : capacity[i, j] <> 0 \}:
  \( \frac{(\theta_i - \theta_j)}{weight[i, j]} \cdot m\_factor, \leq capacity[i, j]; \)

# Minimum flow rate
subject to flowcapmin \{ i in nodes, j in nodes : capacity[i, j] <> 0 \}:
  \( \frac{(\theta_i - \theta_j)}{weight[i, j]} \cdot m\_factor, \geq -capacity[i, j]; \)

# Flow conservation on an inner node
subject to flowcons \{ i in inner \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, = production[i];

# Traditional production constraint
subject to genproduction \{ i in tgen \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, = production[i];

# Min production value storage
subject to flowfromstorageDischarging \{ i in storage \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, = flowmaxdischarge[i];

# Max production value storage
subject to flowfromstorageCharging \{ i in storage \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, \geq flowmaxcharge[i];

# Renewable production constraint
subject to setRewProduction \{ i in rgen \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, <= rewproduction[i];

# Renewable production constraint
subject to minRewProduction \{ i in rgen \}:
  sum \{ j in nodes : capacity[i, j] <> 0 \} ((\theta_i - \theta_j)) / weight[i, j] \cdot m\_factor, = rewproduction[i];
\[-\text{theta}[j]/\text{weight}[i,j]\] * m\_factor, >= 0;

# The amount of energy send to a consumer should be lower or equal to the load of the consumer subject to loadfix \{ i in consumers \}:
\[
\sum \{ j \ in \ nodes : \ capacity[j,i] <> 0 \} \ ((\text{theta}[j] - \text{theta}[i]) / \text{weight}[j,i]) * m\_factor, <= \text{loads}[i];
\]

# Flow to a consumer can never be lower than zero subject to loadMinValue \{ i in consumers \}:
\[
\sum j \ in \ nodes : capacity[j,i] <> 0 \ ((\text{theta}[j] - \text{theta}[i]) / \text{weight}[j,i]) * m\_factor, >= 0;
\]

# Balance supply and demand of energy. subject to prodloadeq:
\[
\sum \{ i \ in \ (\text{rgen union tgen union storage}), j \ in \ nodes : \ capacity[i,j] <> 0 \} \ ((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor, = \sum \{ i \ in \ consumers, j \ in \ nodes : capacity[j,i] <> 0 \} \ ((\text{theta}[j] - \text{theta}[i]) / \text{weight}[j,i]) * m\_factor;
\]

# go ahead and solve the model solve:
# display \{ i in nodes, j in nodes : capacity[i,j] <> 0 \}: i, j, ((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor;
printf \{ i in storage, j in nodes : capacity[i,j] <> 0 \} : 
"%d %d %.3f\n", i, j, ((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor > "update.txt";

# For consumer nodes, generator nodes, renewable generator nodes printf \{ i in nodes, j in nodes : capacity[i,j] <> 0 \} : "%d,%d,%.4f,%.4f,%.4f\n", i, j, ((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor, (abs(((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor) / capacity[i,j]) * 100, (abs(((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor)) / \text{totload}) * 100 > "sol" & outname & ".txt";
printf : "\n" >> "sol" & outname & ".txt";

# Renewable energy curtailment printf \{ i in rgen \} : "R,%d,%.4f\n", i, (sum\{ j in nodes : capacity[i,j] <> 0 \} ((\text{theta}[i] - \text{theta}[j]) / \text{weight}[i,j]) * m\_factor) >> "sol" & outname & ".txt";
printf : "\n" >> "sol" & outname & ".txt";

103
# Traditional generators
printf { i in tgen } : "TG,%d, %.4f \n", i, (sum { j in nodes : capacity[i,j] <> 0 } ((theta[i]-theta[j])/weight[i,j]) * m_factor) >> "sol" & outname & ".txt";
printf : \"n\" >> "sol" & outname & ".txt";

# Consumers
printf { i in nodes, j in consumers : capacity[i,j] <> 0 } : "C,%d, %.4f \n", j, ((theta[i]-theta[j])/weight[i,j]) * m_factor >> "sol" & outname & ".txt";
printf : \"n\" >> "sol" & outname & ".txt";

# inner nodes
printf { i in inner } : "I,%d, %.4f \n", i, (sum { j in nodes : capacity[i,j] <> 0 } ((theta[i]-theta[j])/weight[i,j]) * m_factor) >> "sol" & outname & ".txt";
printf : \"n\" >> "sol" & outname & ".txt";

printf : "\n" >> "sol" & outname & ".txt";

printf "CURTAILMENT,%.4f \n", (sum { i in rgen })
    (rprodmax[i] - (sum { j in nodes : capacity[i,j] <> 0 } ((theta[i]-theta[j])/weight[i,j]) * m_factor)) >> "sol" & outname & ".txt";

printf "EENS,%.4f \n", (sum { i in consumers })
    (loads[i] - (sum { j in nodes : capacity[i,j] <> 0 } ((theta[i]-theta[j])/weight[i,j]) * m_factor)) >> "sol" & outname & ".txt";

printf : \"n\" >> "sol" & outname & ".txt";

# prints flow in i \n
flow in [ %d , %d ] = %.6f \n", i, ((theta[i] - theta[i])/weight[j,i]) * m_factor;

printf { i in nodes } : "theta %d = %.6f\n", i, theta[i]
    >> "sol" & outname & ".txt";

# prints theta and the load for debugging purposes
printf { i in consumers } : "Result theta: %d loads: %d \n
", sum { j in nodes : capacity[j,i] <> 0 } ((theta[j]-theta[i])/weight[j,i]) * m_factor, loads[i];
\#printf \{i in tgen, j in nodes : capacity[i,j] <> 0\} : "
  Theta: %.4f, %.4f \n". theta[i], theta[j];
end;
Appendix B

Data

B.1 Price profiles

**Price Profile 1 - 15%:**

Investment cost storage: \([2900000.00000001, 13950000.00000006, 27910000.00000001, 41860000.00000001, 56070000.00000002, 70020000.00000002, 83980000.00000002, 98180000.00000002, 112140000.00000005, 126090000.00000005, 140300000.00000005] \)

cost EENS: \([3528474469.7494464, 3107636031.9500046, 2784614807.7014813, 2789613553.357315, 2802592119.3286557, 2363665554.1219077, 2735279979.726065, 2445546334.3714484, 2450713594.8968477, 2129830116.1149595, 2257281809.4322143] \)

RoI: \([0, 420838437.7994418, 743859062.047965, 738861316.3921313, 725882530.4207907, 1164850815.6275387, 79319490.0233812, 1082928135.376298, 1077760874.8525987, 1398644353.634487, 1271192660.3172321] \)

RoI percentage: \([0.0, 3.0167629949780763, 7.388613163921313, 7.258825304207907, 1.1648508156275387, 0.793194900233812, 1.082928135376298, 1.0777608748525987, 1.398644353634487, 1.2711926603172321] \)

**Price Profile 1 - 30%:**

Investment cost storage: \([2900000.00000001, 120600000.00000001, 240650000.00000002, 360600000.00000002, 480650000.00000002, 600600000.00000002, 720650000.00000002, 840600000.00000002, 960650000.00000002, 1080600000.00000002, 1200650000.00000002] \)

cost EENS: \([3528474469.7494464, 3612486017.335702, 2679600542.665202, 2260382248.0610847, 2425684029.702987, 2471568463.829943, 1794189717.182578, 1944414583.345976, 2305762903.009005, 2338546855.457725, 2500746538.124537] \)

RoI: \([-0.84011547.58625558, 848783927.0842443, 1268092221.6883616, 1102794440.0464592, 1036906005.9195032, 1734248752.5688676, 1584059886.4034703, 1222711476.7455416, 1189927614.2917213, 1027727931.6249094] \)
Price Profile 2 - 15%:
Investment cost storage: [6000000.000000002, 293750000.0000001, 587750000.0000002, 881500000.0000004, 1180500000.0000005, 1474500000.0000007, 1768500000.0000009, 2067000000.0000011, 2361000000.0000014, 2654750000.0000017, 2953750000.0000020]
cost EENS: [3528474469.7494464, 3107636031.9500046, 2784614807.7014813, 2735279079.72065, 2445546334.3731484, 235713594.9068477, 2129830116.1149595, 2257281809.4322143]
RoI: [0, 420838437.7994418, 74359662.047965, 73861316.392131, 72588235.376298, 1077760874.8525987, 1398644353.634487, 1271192660.3172321]
RoI percentage [0.0, 420838437.7994418, 74359662.047965, 73861316.392131, 72588235.376298, 1077760874.8525987, 1398644353.634487, 1271192660.3172321]
Price Profile 2 - 30%:
Investment cost storage: [9500000.000000004, 475500000.0000002, 955000000.0000004, 1435000000.0000007, 1918500000.000001, 2365000000.0000014, 2874000000.0000017, 3390000000.0000020, 3837000000.0000023, 4314500000.0000026, 4800000000.0000029]
cost EENS: [3528474469.7494464, 3107636031.9500046, 2784614807.7014813, 2735279079.720659, 2445546334.3731484, 235713594.9068477, 2129830116.1149595, 2257281809.4322143]
RoI: [0, -84011547.58625555, 848739270.842443, 1268092221.688316, 1102790440.0464592, 105699005.9195032, 1734284752.5668676, 1584059886.4034703, 1227711476.7455416, 1189927614.2917223, 1027729316.6246094]
RoI percentage [0.0, -84011547.58625555, 848739270.842443, 1268092221.688316, 1102790440.0464592, 105699005.9195032, 1734284752.5668676, 1584059886.4034703, 1227711476.7455416, 1189927614.2917223, 1027729316.6246094]
Price Profile 3 - 15%:
Investment cost storage: [9500000.000000004, 475500000.0000002, 955000000.0000004, 1435000000.0000007, 1918500000.000001, 2365000000.0000014, 2874000000.0000017, 3390000000.0000020, 3837000000.0000023, 4314500000.0000026, 4800000000.0000029]
cost EENS: [3528474469.7494464, 3107636031.9500046, 2784614807.7014813, 2735279079.720659, 2445546334.3731484, 235713594.9068477, 2129830116.1149595, 2257281809.4322143]
RoI: [0, 420838437.7994418, 74359662.047965, 73861316.392131, 72588235.376298, 1077760874.8525987, 1398644353.634487, 1271192660.3172321]
RoI percentage [0.0, 0.8813370425119197, 0.7785030476692463, 0.5156045473776212, 0.3783531739420925, 0.4861472936675869, 0.275989732088591, 0.3223959914785048, 0.2808863369436013, 0.32417298728345956, 0.2648318042327566]

Price Profile 3 - 30%:

Investment cost storage: [9500000.000000004, 807500000.0000002, 1623000000.0000005, 2430500000.0000001, 3249000000.0000005, 4061000000.0000002, 4869000000.0000002, 5684000000.0000002, 6499500000.0000002, 7307000000.0000002, 8122500000.0000004]

cost EENS: [3528474469.7494464, 3612486017.335702, 267990542.663202, 2260382248.0610847, 2425684029.702987, 2471568463.829943, 1794189717.1825788, 1944414583.345976, 2305762993.003905, 2338546855.457725, 2500746538.124537]

RoI: [0, -84011547.58625555, 848783927.0842443, 1268092221.6883616, 1102790440.0464592, 1056900605.9195032, 1734284752.5668676, 1584059886.4034703, 1222711476.7455416, 1189927614.2917213, 102772931.6240094]

RoI percentage [0.0, -0.84011547.58625555, 0.848783927.0842443, 0.1268092221.6883616, 0.1102790440.0464592, 0.1056900605.9195032, 0.1734284752.5668676, 0.1584059886.4034703, 0.1222711476.7455416, 0.1189927614.2917213, 0.102772931.6240094]