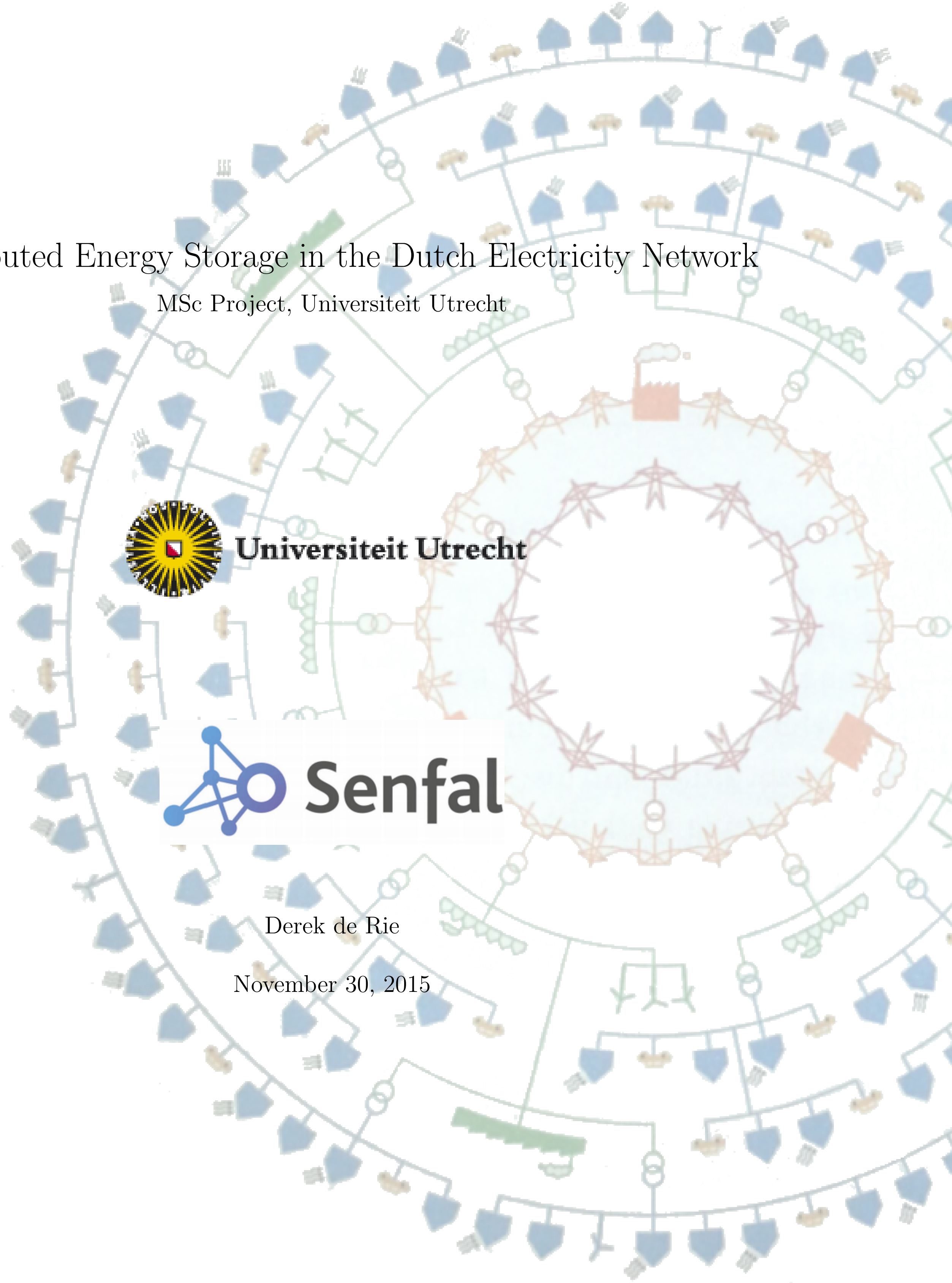


Distributed Energy Storage in the Dutch Electricity Network

MSc Project, Universiteit Utrecht

Derek de Rie

November 30, 2015

Abstract

Renewable energy sources and shifting electricity demand patterns present challenges to electricity grid operators. Large batteries could provide a solution to grid damage. However, previous studies have shown that costs of batteries are currently too high. Here, we investigate the profitability of storage solutions when energy trading is added to the decision control. This report features an extensive overview of the Dutch electricity system, including various electricity markets. The topic is researched both as an optimisation problem as well as a profitability study in the current Dutch situation. This profitability is tested using a custom simulation model that features a realistic cable damage model, various sources of Dutch electricity demand data, realistic forecasting and two main electricity markets. We evaluate numerous strategies using this model. As part of this evaluation we investigate how local search can be used to find solutions. We discuss the boundaries to which our model can be extended, and provide a decision tool for grid operators to answer a question of growing importance: Is energy storage a preferred and profitable solution to electricity grid damage?

Acknowledgements

I would like to thank my supervisor Marjan van den Akker for her support, critical questioning and helpful insights during my thesis project. Under her guidance I was able to freely explore many of my own (sometimes-not-so-bright) ideas. This allowed me to approach the topic of this report in many different ways, and bring the project to its current state. I'm grateful for the trust and patience I received while undertaking this challenge.

I would also like to thank Hubert Spruijt, Steven Reiz and Sander ten Kate, my wonderful colleagues at Senfal. Senfal has allowed me to combine my curiosity for optimisation problems with a real-world focus. I would like to express special thanks to Hubert Spruijt for his patience in explaining me various concepts regarding electricity or the electricity system.

Contents

Chapter 1

Introduction

Electricity grids are subject to major technological developments and changes. In distribution grids, Distributed System Operators (DSOs) are dealing with an increasing number of powerful appliances and uncontrollable Renewable Energy Sources (RES). As a result, there is an ongoing effort in reducing peak usage in these grids. Furthermore, with the increase in renewable energy sources, there is a growing gap between energy availability and energy demand. In order to maintain stable grids and to reliably provide electricity, research has focused on two main methods: demand-response (DR) and distributed energy storage (DES).

These two methods both represent a fundamentally different view on these issues. Demand-response matching entails adapting the electricity usage to the availability (production) of energy. For example, an electrical heat-pump that turns on when nearby RES (such as a wind-turbines or Photo Voltaic (PV) installations) produce energy. Of course, this requires the appliance to be flexible to a certain extend. Appliances that are required to turn on immediately at the time the user requests (for example televisions or kitchen appliances) are not suitable for demand-response matching.

Incorporating DES in the electricity grid has the benefit that abundant electricity can be stored and used at a later time. For example, a household with installed PV could save up the electricity during the day (low demand), and use the electricity in the evening (high demand). This removes the need for switching the usage pattern, but results in a loss of energy due to imperfect efficiency of the storage unit.

Both DR and DES are capable of reducing peak usage in grids. Peak usage refers to short periods where there is a large electricity flow through the power grid. These peaks may cause damage to the grid (overheating cables) and connection utilities (e.g. electrical transformer). Traditionally, DSOs have resolved these issues by replacing parts of the grid to satisfy the peak usage. But as electricity peaks are becoming more common, grid reinforcements are becoming a substantially larger operating cost. Thus, there is growing interest in solving these peak-related issues without traditional grid enforcements. Both DR and DES can lower electricity peaks. This is referred to as peak-shaving. Furthermore, by limiting the interaction of prosumers (joint consumer and producer of electricity) with the grid, both methods can reduce electricity costs. Here, we focus on using storage (DES) for both these tasks.

Although peak-shaving in electricity grids is discussed in scientific literature for more than a decade, Bar-Noy et al. [1] first discusses storage-facilitated peak-shaving as a mathematical problem. They provide a number of properties and algorithms for both the offline (full knowledge of near future data) and online (limited knowledge of the problem) case. Although their models of storage units are basic, they show that peak-shaving using storage is an optimisation problem, and that peak-shaving can be achieved by introducing storage units.

Currently, DES remains to be widely implemented in electricity grids. The main factor remains the high cost of batteries, flywheels and other storage technology. Nykamp et al. [2] conducted a break-even analysis for storing PV electricity locally. The authors note that introducing storage in low-voltage grids can reduce the need for traditional grid enforcements (cable replacement) in the distribution grid, lower feed-in peaks from upstream grids and save money as storage owners do not need to purchase additional energy when there is no RES production. Although the authors ultimately conclude that investing in storage assets is only profitable under "*extremely positive conditions*" (high cable costs and long storage lifetime), they do note that there another way in which investments can be more profitable. By exploiting the price spread (arbitrage) on electricity markets while maintaining grid balance, another mode of income can be utilised. If a storage unit is not only used for peak-shaving, the problem inherently becomes multi-objective.

This notion of combined grid protection and arbitrage by electricity storage is further discussed in Ramezani et al. [3], which was published around the same time as Nykamp et al. [2]. Similar to Bar-Noy et al. [1], the authors approach the problem both in an offline and online setting. The authors incorporate electricity market prices to maximise profit (arbitrage), while keeping the cable usage within allowed limits. Ramezani et al. [3] extend the simple grid structure of earlier publications, and also focus on the economic losses due to cable damage. The authors mention that "*There is currently no consensus among engineering experts about a standard cost function that represents economic losses experienced by overheating a cable*". In their simulations economic losses are scaled linearly with the energy that exceeds a set limit for the cable. Ultimately, the authors show that electricity storage could present a profitable and realistic solution to harmful peaks in distribution grids.

The idea of a multi-objective optimisation is further explored in Nykamp et al. [4]. The peak-shaving objective is taken as a constraint (keep the system between the allowed values), while the objective becomes to maximise profit on electricity markets. The authors use an offline linear programming algorithm to calculate the profits of a multi-objective approach. The assumption is made that electricity prices are known, while in reality these have to be predicted. The results are promising, but whether the same performances can be acquired using realistic forecasting precision is not yet known. We therefore include forecasting in our evaluations of algorithmic approaches.

Finally, Nykamp et al. [4] recommend two ways through which storage units can be incorporated for grid enforcements. Either DSOs should be allowed to install and operate storage units themselves (this is currently prohibited by law), or DSOs should be allowed to reimburse electricity (trading) companies for operating storage units for peak-shaving on their behalf. These business models are further discussed in Section 2.5.

In this thesis we attempt to combine and extend the exploratory efforts regarding electricity storage for grid reinforcement and energy trading. Our goal is to extend the current work towards real-world implementation. Therefore, we need to develop algorithms and evaluations that deal with more complex situations and cost functions. In short, we will:

1. Follow a completely algorithmic approach. Economic and econometric studies on the subject of storage in electricity grids have relied on assumption how much profit can be achieved by implementing smart strategies and forecasting models. For instance, Nykamp et al. [2] presented an economic approach to calculate break-event points for the cost of storage systems that both prevent grid damages as well store solar energy. By using actual algorithms, strategies and forecasting models this study removes the need for assumptions on the valuation of storing electricity. Furthermore, these algorithms such be usable in practice.

2. Evaluate algorithms and strategies in an online evaluation. Offline calculations such as presented in Nykamp et al. [4] do not incorporate changing conditions as well as unexpected events. We believe that by assessing online algorithms our findings will be more suitable for real-world implementation.

3. Incorporate realistic forecasts. In Ramezani et al. [3], a simple electricity demand forecast is made by averaging a daily pattern. Many advances in forecasting electricity production, demand and market prices have been made. We believe that evaluating algorithms using realistic and dynamic forecasts will result in a better view on the possible profitability of storage units.

4. Provide a more advanced model for cable damage. A concern of the authors of Ramezani et al. [3] was that no consensus model for cable damage exists. In a later paper, the same lab showed that brief

interruptions have a relatively large positive effect on cable damage [5]. We therefore extend the cable models used in the literature, to more closely evaluate the economic costs of cable damage, with or without nearby storage units. Although our approach still does not approach the level of complexity advanced physics-based cable heating models, we do extend the linear cable damage model assumed by all publication discussed above.

5. Solely evaluate algorithms on real-world data from the Netherlands. An effort was made to collect relevant data. We believe that the profitability of storage units is greatly affected by the electricity markets and their price spreads. The conclusions and recommendations in this thesis therefore apply mainly to the Netherlands.

In this thesis, the words *storage unit* and *battery* can be used interchangeably. We think that battery is a more intuitive concept for later chapters. However, the algorithms and strategies in this thesis can be applied to other forms of electricity storage as well.

Senfal

This study was carried out in a combined effort between Utrecht University and Senfal. Senfal is a start-up company aiming to play an active role in balancing electricity grids. By implementing both demand-response matching and distributed energy storage, Senfal leads a joint effort by consumers, producers and grid operators to prepare electricity grids for the future. The aims, activities and role in this project by Senfal are introduced in Section 2.5.

Thesis Outline

In this thesis, the multi-objective storage problem is approached both as a fundamental mathematical problem, as well as science relating to a real-world issue. The report therefore is structured in two parts. The first serves as an introduction into electricity grids, policies, technological development and other relevant information. The second part of the report then focus on the algorithms and analysis applied to the electricity grids.

First, we provide an extensive overview of the electricity sector, electricity markets and ongoing relevant developments in Chapter 2. Here, we also provide an overview of the quality of forecasting models present today. In Chapter 3 discusses how electricity cables, batteries and other entities are modelled in this study. The parameter-based models for cables and batteries are introduced, as well as of all the data that models electricity producers and consumers. Chapter 4 serves as a bridge from the electricity grids into the field of optimisation. First, we explain the concept of *grid enhancement* through electricity storage. Then, we discuss relevant literature, including the publications mentioned in this introduction. Finally, we demonstrate the peak-shaving capabilities of storage units using a dynamic programming algorithm. We extend the problem definition in Chapter 5 by adding energy trading (variable energy prices) in the optimisation problem. We then provide algorithms that can be used to solve these problems. Chapter 6 introduces our online simulation model to evaluate numerous approaches to the optimisation problem. In Chapter 7 we discuss and analyse the need for a local-search on parameter settings to further optimise the strategies. In Chapter 8 we apply the best performing algorithms from earlier chapters to real-world situations, and revisit the investment decisions discussed by Nykamp et al. [2]. Chapter 9 summarises our main efforts, and offers some recommendations for stakeholders in the grid stabilisation issue.

Chapter 2

Background

Contents

Energy markets, systems and technologies are rapidly changing on a global scale. In the Netherlands and other European countries, there is a steady rise in electricity production from small, distributed energy resources such as wind and solar energy. These energy sources are scattered abundantly throughout the Dutch electricity network, and individually produce less power than traditional energy resources such as coal, gas or nuclear power plants. The introduction of more and more of these Distributed Energy Resources (DERs) creates a challenge for the companies in control of the Dutch electricity network. With less predictability (e.g. weather circumstances) among more resources, an effort is required to keep the Dutch electricity grid balanced and reliable.

In the following chapter we provide an overview of policies, developments and research findings related to the field of electricity. In Section 2.1, an overview is provided on which parties mainly operate in the system. Section 2.2 describes which are the main markets for electricity. Development and expected development concerning the grid are summarised in Section 2.3. This study incorporates forecasting of electricity use and electricity prices. In Section 2.4 we therefore list recent efforts in forecasting electricity use and prices. Finally, Section 2.5 introduces Senfal, the company that this project is done in partnership with.

2.1 The Dutch Electricity System

Electricity is an essential part of our society. In order to make sure there is an affordable and reliable availability of electricity, the Dutch electricity system consists of different institutions. To understand what the role of each of these institutions is, we will first introduce the structure of the electricity grid.

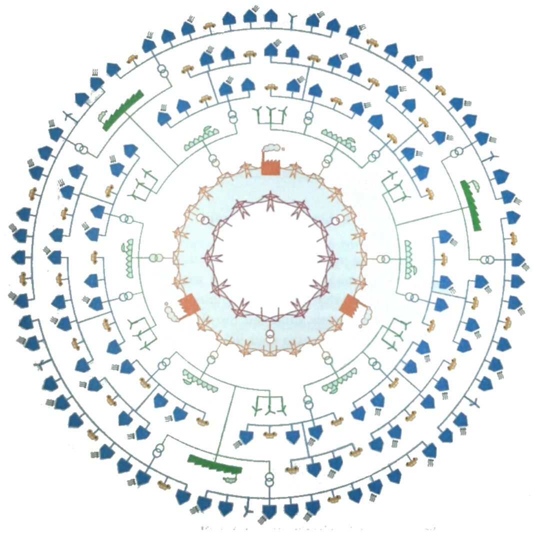


Figure 2.1: Functional illustration of the 4 levels in the Dutch electricity grid. *Red*: Transmission Grid, *Orange*: Sub-transmissiont Grid , *Green*: Regional Distribution Grid ($\sim$ 100 MVA), *Blue*: Local Distribution Grid. "*Elektriciteitsnet, ingedeeld naar functies*" van Oirsouw and Cobben [6]

The purpose of the electricity grid is to facilitate transportation of electricity from an international level to the local level at individual connections. In order to achieve this, all local, regional and national grids are connected. An important distinction is made between transport and distribution grids. The transport grid facilitates the large-scale transport of electricity (including international transport), in order to supply each region with enough electricity. The distribution grid then distributes this bulk of electricity to all the individual connections [1]. [6]

Figure 2.1 illustrates the different levels of electricity grids. At the highest level (inner circle), the transmission grid (1, *red*) connects the Dutch grid with neighbouring electricity grids. Additionally, this grid facilitates the national transport of power generated by large power plants. The sub-transmission grid (2, *orange*) connects the transmission grid to the distribution net. Smaller power plants and large industries are also connected to this grid. The regional distribution grid (3, *green*), connects industries, wind-turbines and thermoelectric power plants to both the sub-transmission grid and the local distribution grid. The local distribution grid (4, *blue*) connects all households, distributed producers and smaller electricity consumer to the grid.

2.1.1 System Operators

The grid is maintained by system operators. The Transmission System Operator (TSO) is an entity responsible for the transportation of electricity at the transmission and sub-transmission grid level. In the Netherlands there is one TSO, TenneT. TenneT maintains this power grid, and monitors and regulates the total production and consumption of electricity. The TenneT Imbalance market (Section 2.2.3) is therefore managed by TenneT.

Aside from the TSO, a DSO is entrusted with maintaining the distribution grids (both regional and local). In the Netherlands there are a number of DSOs, each responsible for one or more regions. The TSO and DSOs together are responsible for facilitating the transportation of electricity throughout the entire grid. By law, these entities are required to connect everybody that requests a connection to the grid. Since

[1] Connections here can be residential houses, companies, office building, wind turbines, solar facilities, etc.

Name	Time Scale	Price Range (per MWh)	Main purpose
APX Day Ahead	1 hour (24h adv.)	20-80	Main wholesale electricity market (prediction based)
APX Intra Day	15 min.	20-80	Adjusting predictions and bid-imbalances on APX Day Ahead market
TenneT Imbalance	1 min.	(-50)-200	Restoring system imbalance

Table 2.1: Overview of the main Dutch electricity markets

there is a strict assignment for each part of the grid to one entity (transmission/transport grid to TSO, each distribution grid region to a DSO), each cable, transformer or other facility has a direct controlling entity.

2.2 Electricity Markets

The Dutch electricity market is organized in different markets, characterizable by purpose, quantities, time scales and price ranges. In a simplistic view, the market consists of producers and consumers of electricity. If a party were to produce electricity in the future (by any means), they can sell a contract for that energy. Likewise, consumers of energy have to acquire these contracts to be able to consume this energy. To facilitate this trade in contracts, a number of markets exist.

Parties are not required to participate in any of these markets to be able to legally produce or consume electricity from the national grid. When two parties agree to a contract themselves, it's only necessary to inform the TSO of such a contract. These kinds of contracts are usually made long-term periods in the future. For shorter periods, or periods not far in the future, the electricity markets form excellent places to find a buyer or seller of electricity. The TSO must be notified of these transactions.

In case a party does not behave according to what has been notified to the TSO, the electricity grid might become unstable. If a party for instance uses 20MW instead of 10MW (which was notified to the TSO) during a 1-hour period, a shortage of electricity might arise. In order to prevent this, the TSO maintains an imbalance price. For all energy consumed or produced that does not conform the notified contracts, the TSO charges this imbalance price for the deviated amount of energy. In the previous example, the party will be charged 10 MWh of electricity against the imbalance price. By updating this price real-time, the TSO can indirectly resolve grid imbalances. A high imbalance price will cause parties to produce more, or consume less energy. Thus, there is an incentive to all parties in the system to negate the overall grid imbalance. The TSO and all connected parties in the system therefore prevent grid imbalances as a whole. Note that this system is different compared to for instance Germany, where all parties are incentivised to prevent their own imbalance [7].

Below we will present an overview of the electricity markets. The main public electricity market is the APX Day-Ahead market. On this market, contracts are formed using auctions. The APX Intraday market allows parties to exchange contracts up to The main public electricity market is the APX Day-Ahead market. On this wholesale market, both energy producers and consumers place energy bids on different hour-slots, 24 hours in advance. Up to 15-minutes before the contracts take place. Additionally, the TenneT Imbalance market features a per-minute price to prevent imbalances in the network. An overview of these markets is provided in Table 2.1. In addition to these markets, a number of other separate and markets, such as the TenneT Primary Reserve, exists. These markets were not included as the markets are not applicable to the scope and size of the distributed energy storage topic. In addition, there is limited availability on the transactions in these markets.

2.2.1 APX Day-Ahead market

The APX day-ahead market, from now on referred to as the APX market, works using a bidding system. This bidding system requires parties to submit their *bids* of energy demand or supply at 10:00 am on the day before for a 24-hour period. According to the hour of day, these bids are therefore made 14 to 38 hour in advanced [2]. The prices is established by using a double sided blind auction system. The bidding system works by collecting all the bids, sorting the supply and demand bids separately, and finding the price for which an equal volume of supply with higher or equal bidding price matches a demand volume with lower or equal price. This equilibrium price is the price each party within those volumes pays or receives for their share of energy.

2.2.2 APX Intraday market

The APX Intraday market is an online exchange of energy contracts. Energy contracts can be offered at the market at a certain price. Parties can contact the party that offers such a contract in order to finalise a deal. The APX Intraday market, in contrast to the APX Day-ahead market allows specific exchanges between parties. Currently, it's not possible for automated participation in the market. All transactions have to be manually done.

2.2.3 TenneT Imbalance market

The TenneT Imbalance Market is used to give incentives to parties to restore electricity imbalance. Electricity imbalance occurs when the supply of energy is higher or lower than the demand. This causes the frequency of the alternating current to change, potentially resulting in damage to assets or even blackouts. The imbalance price is determined by the TSO, and stimulates parties to up- or down-regulate their production/consumption. When a party causes electricity imbalance by not acting according to their APX-day ahead, APX-intra day or other agreements known to the TSO, the imbalance price is charged.

In the Netherlands, the collection of markets is aimed towards easily allowing parties to partake in the electricity trade, while giving incentives to all parties to keep the system stable. This is in contrast to for instance Germany, where incentives are in place to keep parties from causing system imbalance, even it that works against the system imbalance [7]. As shown in Figure 2.2, this causes the imbalance price in the Netherlands to be much more responding to electricity imbalance. Furthermore, their was less overall electricity imbalance in the Dutch grid.

2.3 Developments

In an effort to reduce the harmful effects of carbon-based electricity production, countries are actively trying to shift their energy production to renewable energy sources (RES). In Europe, members of the European Union have agreed to achieve a 20% renewable energy generation by 2020[3]. In the Netherlands, Germany and other European countries the main types of renewable energy are biomass-, solar- and wind-energy. One of the fundamental differences with traditional energy sources such as coal and gas is that these new energy sources are typically distributed throughout the electricity grid. Figure 2.1 schematically shows how wind and solar energy are found in the distribution level grid, while larger, traditional power plants are found in the transport and transmission grids. In the Netherlands, 10,1% of the total electricity generation came from renewable energy sources. The Dutch government has installed a policy to increase this percentage to 16% in 2023[4].

Besides a change in the generation of electricity, the energy consumption is also subject to developments. Over the past decades, electrical devices have required more and more energy to operate. Additionally, electric vehicles have become available to a larger audience. In the Netherlands, predictions on the amount

[2] https://www.apxgroup.com/trading-clearing/auction/
[3] https://ec.europa.eu/energy/en/topics/renewable-energy
[4] https://www.energievergelijken.nl/nl/energiemarkt/energieproductie-in-nederland

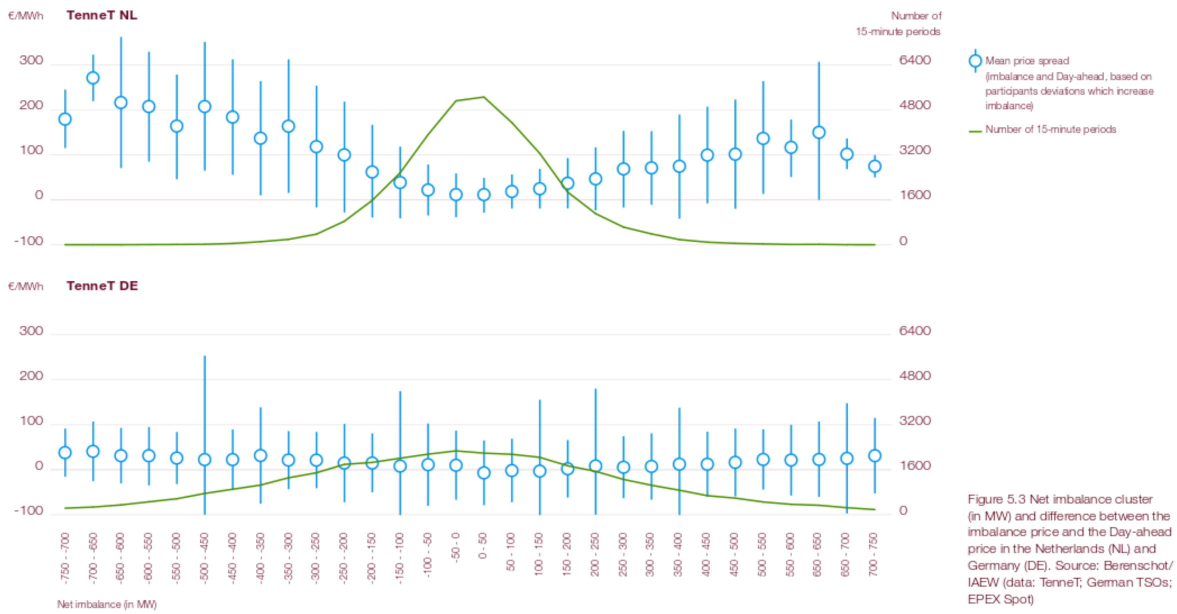


Figure 2.2: "*Net imbalance cluster (in MW) and difference between the imbalance price and the Day-ahead price in the Netherlands (NL) and Germany (DE). Source: Berenschot/IAEW (data: TenneT; German TSOs; EPEX Spot)*" [7]

of electric vehicles in 2020 range between 200,000 and 1 million. Although the specific electricity consumption depends on EV type, style of drive and distance, the average daily electricity consumption of EVs is more than 66% of the daily average energy consumption of a Dutch household. Adding EVs to a residential area greatly affects the energy demand on the grid (Figure 2.3) [6].

Demand-Response and Distributed Energy Storage Both the increasing energy demand and energy generation in distribution grids could result in large energy flows through the distribution grids. A demand that is too large for the cable dimension results in permanent damage to the grid. As eluded to in the introduction, there are 2 main approaches to this problem. By shifting the energy consumption to the availability of generated electricity, the amount of electricity that has to be transported from and to the distribution grid can be lowered. This is called Demand-Response. A cooling machine (e.g. refrigerator) that turns on when there is electricity available is an example of demand-response. Similarly, electric vehicles could be charged when during times of distributed solar- or wind-energy generation. Naturally, not all demand can be shifted towards periods of energy availability. Many household appliances, such as lights, television and cooking devices are turned on when people use them. Similarly, industrial energy usage is dictated by office hours, availability of resources and process-specific requirements.

The second approach to deal with these developments is distributed energy storage. Instead of generating electricity (using solar- or wind-energy), transporting the electricity to the transport- or transmission grid, and transporting energy back during a later time for consumption, electricity could also be stored. How energy storage can prevent damage to cable grids in further discussed in Chapter 4.

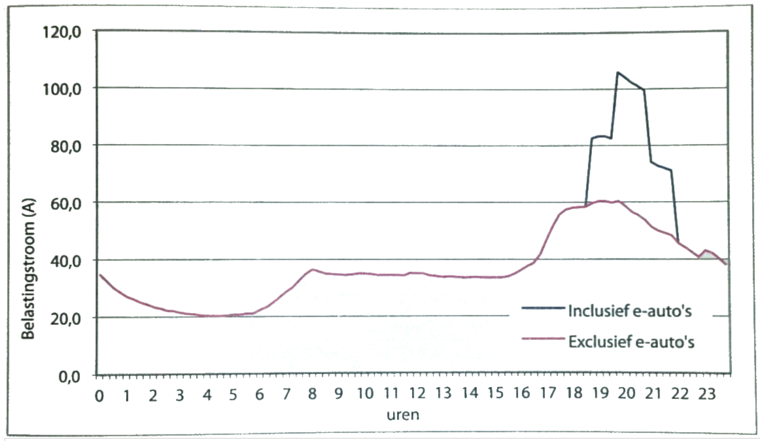


Figure 2.3: Example of electricity demand with ("*Inclusief*") and without ("*Exclusief*") electric vehicles in a typical Dutch residential area consisting of 40 households and 10 electric vehicles. [6]

2.4 Forecasting

Over the last decades there have tremendous developments in the electricity business. Driven by economic, political, technological and environmental factors, there is rapid, ongoing change in the way electricity is produced, consumed and traded. In many of these developments, there is a rising need for accurate forecasting models, to facilitate planning (e.g. demand-response mechanisms), financial optimisations (e.g. price forecasting) and necessary grid control (e.g. preventing harmful peak usage). This section provides an overview of publications on forecasting electricity demand (consumption), market prices and electricity generation.

2.4.1 Load Forecasting

Load forecasting is becoming an important discipline for decision makers in the electricity sector. From demand-response planning to trends that affect grid usage, both short- and long-term forecast models are of great interest. Hahn et al. [8] provides an overview of the relevance of different forecasting horizons. Short-term load forecasts (STLF) typically predict loads up to one-week ahead. Medium-term load forecasts (MTLF) range from one week to one year, while long-term load forecasts (LTLF) have forecast horizons of over one year. Day-to-day operations are mostly dependent on STLF. In this study, we restrict our scope to STLF [8, 9].

Campillo et al. [10], Alfares and Nazeeruddin [11] and Hahn et al. [8] give a good reflection on the types of models used in load demand forecasting. Campillo et al. [10] distinguishes between the following methodologies:

1. Trend. This method simply explains the load demand as a function of the time, and exclusively uses historical data. Example techniques in this methodology are multiple regression, exponential smoothing, stochastic time series and iterated weighted least squares. Alfares and Nazeeruddin [11] provides a comparison of these methods.

2. Similar Day. This methodology uses historical data to find earlier situations on which the forecast is based. Typically these methods take into account the day of the week, weather information and seasonality. This is a common methodology for forecasting, for which many techniques exist. Hahn et al. [8] presents an overview of the implementation of regression models, time-series approaches, neural networks, support vector machines and hybrid approaches that fall within this methodology.

3. End-Use. End-Use models are built using a bottom-up approach. The underlying assumption here is that electricity demand can be derived from properties of the situation. For residential load demand, a model could be constructed from the number of residents, their ages, the types of equipment present etc. These models allow for long-time forecasting, require little or none historical data, but do require extensive knowledge of the situational properties.

4. Econometric. Econometric models "*combine economic theory and statistical analysis for forecasting electricity demand, by establishing the relationships between energy consumption and the factors that influence it.*" [10]. These models require extensive sample size, since they reflect heavily on extrapolation.

Overall, the extensive amount of publications here suggest that the choice of model is heavily dependent on the required forecasting horizon, the amount of data present and the extend of the insights on the situation [10, 11, 8]. Likewise, the effectiveness of these models will not be equal when comparing load demand in different countries.

Nijhuis [12] performed a large study on load forecasting in the Dutch low voltage (LV) distribution grid. The models in this study were used to forecast electricity demand on a 24-hour forecast horizon. The models were typical Similar Day models, which used time-series and regression techniques. The study incorporated weather data, although it was shown that these variable were of modest effectiveness. The Nijhuis [12] publication presents a valuable resource on forecasting quality of Dutch electricity demand data. According to the report, demand data can be forecasted at a precision around MASE=0.7. The MASE statistic and implications of the forecasting quality measures are discussed in Section 6.2.5.

Asare-Bediako et al. [13] utilised artificial neural networks to forecast electricity demand. Data was obtained using smart meters. Using only historical data, the study showed that precise forecasts were formed. Furthermore, the results showed that the necessary amount of training data was not very large, and that adding large volumes of training data did little to nothing to the forecasting precision.

Kadurek et al. [14] presented a characterization of residential load demand in Dutch low voltage (LV) networks. Although their method enables load profile generation, the predictive power of these powers is not assessed.

2.4.2 Market Price Forecasting

Internationally, substantial research has been done on forecasting electricity prices. For instance, Lu et al. [15] focus on the prediction of price spikes, applied to the National Energy Market (NEM, Australia) spot prices. They find that the demand-supply relationship is the main factor affecting the market price. Amjady and Keynia [16] also use data-driven approaches to predict price spikes, using probabilistic neural networks. They successfully evaluate their model on the American PJM (Pennsylvania–New Jersey–Maryland) electricity market. Their study is mainly an extension of the works of Pao [17], who show how artificial neural networks successfully model and forecast electricity market prices of the European Energy Exchange (EEX).

Another study that examines electricity market spot prices is performed by Mazengia and Tuan [18]. Evaluating their results on the Canadian Ontario electricity market, they show that the forecasting errors depend very much on the price volatility. In Canada, electricity prices tend to vary more in summer, due to different power profiles of solar, wind- and water reservoir facilities. They show that during winter and spring their models perform well (MAPE=5%,3%) compared to summer (13%)[5].

Alvarez-Ramirez and Escarela-Perez [19] apply detrended fluctuation and Allan analyses, methods typically used in physics, to Canadian electricity markets as well. In relation to Mazengia and Tuan [18], they find lower correlations in winter between demand and prices as compared to summer. Their analysis show strong cycling patterns with regards to seasons and time.

These studies ubiquitously claim to model and forecast their research markets well, although none of the studies focus on more than one market. It seems that both in method design and result analysis the

[5]Mean Absolute Percentage Error (MAPE), see Nijhuis [12] for detailed explanation

domain-specific knowledge on the market and electricity system is taken into account. It is therefore hard to hypothesise which of these methods perform well on Dutch market data.

With regards to the Dutch hourly electricity market, all publications address the APX day-ahead market. Huisman and Mahieu [20] and Huisman et al. [21] characterise the behaviour of this market. Perhaps surprisingly they show that "*hourly electricity prices do not follow a time series process but are in fact a panel of 24 cross-sectional hours that vary from day to day*" [20]. Together they find that a regime-switching framework is capable of modelling the electricity price, and performs better than consider each hourly price as an mean-reverting process.

Additionally, Boogert and Dupont [22] present a model that explains the probability of a spike in the APX day-ahead market, in relation to the reserve margin on the whole system in the Netherlands. They find that the relatively high temperature (compared to the average temperature at said time) is the best indicator for price spikes.

With regards to later evaluations in this report, these publications cannot be used to quantify the quality of forecasts. Although the publications indicate important indicators and predictors, methods discussion in Chapter 6 cannot use this information. In Chapter 6 we therefore assumed that state-of-the-art APX forecasting was of quality similar to using 24-hour prior APX prices as predictions.

2.5 Senfal: Towards a Balanced, Green Energy System

Senfal is a start-up aimed at introducing distributed energy storage in the Dutch electricity network. Situated in the Utrecht University' UtrechtInc. incubator, tries to show DSOs, Dutch: Netbeheerders van midden- en laagstroom) that investing into energy storage at strategic points in their electricity networks is profitable. Typically, Dutch DSOs are hesitant to experiment with developing their networks. Therefore, Senfal tries to act as a catalyst by developing a strategy for incorporating energy storage in the electricity networks.

In the business model currently developed by Senfal, distributed energy storage will be a service Senfal provides to DSOs. Senfal is responsible for the energy storage itself, and in return guarantees the DSO of a certain amount of electricity at desired times. The energy storages can be charged when too much electricity is available (e.g. on a sunny day due to abundant solar cells), or discharged during peak times.

The storage units can save the DSO money by removing the need for thicker cables (i.e. grid reinforcements). Additionally, by preventing too much or too little electricity at times, cables and equipment are damaged less, resulting in lower replacement costs. In times of total network failure, energy storages can provide neighbouring parts of the network with an emergency supply of energy. This, in turn, reduces damage payments the DSO is forced to pay.

The increased network stability and lower equipment/material consists represent one of the three modes in which this model represents a profitable business model. By Senfal acting as a service provider, the DSO and Senfal can agree to give Senfal permission to use excess electricity for trade. In the Netherlands, DSOs are not allowed to trade electricity on the energy market, as they are legally obliged to maximize the coverage and robustness of the electricity network. Therefore, by including Senfal as a service provider, the model allows for using excess energy for energy trade. In the Netherlands, the Amsterdam Power Exchange (APX) (APX) Day-Ahead and the real time, rapidly changing TenneT Imbalance electricity markets can be entered with the excess electricity. In conclusion, the idea behind Senfal is to provide a service to DSOs in which the DSO gains a more durable, better adapting and more robust electricity network. Additionally, the service allows Senfal to participate in the electricity market. The profits that are made by "trading" electricity using a storage unit can lower the cost of the service.

This project is carried out together with Senfal. Senfal adds domain knowledge, relevant data sets and its contacts in the industry to guarantee that the research is as applicable to the real-world as possible.

2.5.1 Enexis Smart Storage Unit

In 2012, the largest Dutch DSO (Enexis) launched a pilot project to test how large-scale energy storage could relief the distributed electricity grid. This battery was called the Smart Storage Unit (SSU). The battery has a storage capacity of approximately 230 kWh. The maximum discharge rate of the battery is 400 kW, while the maximum charge rate is 100 kW. The battery was placed next to a residential area with an average daily peak load of 385 kWh [23, 24]. In this project, we will use this battery several times as a model for realistic storage size and technical capabilities in residential areas. Currently, we are collaborating with Enexis of applying the algorithms from this study to the SSU. This collaboration is ongoing, and at the time of writing this thesis the algorithms have not yet been implemented on the SSU.

Chapter 3

Modelling Entities in Electricity Networks

Contents

In this study, we discuss how storage units, or batteries, can be used to relieve damage in electricity grids. In this chapter we present models for electricity grids, consisting of cables, batteries and other entities that produce and/or consume energy. From now on, we refer to the latter category as **nodes**. Nodes can refer to every connected entitiy to the electricity grid that produces and/or consumes electricity. Nodes can range from households to entire neighbourhoods, from PV-panels to windmill-parks and from small companies to entire industries.

First we introduce a model that connects these cables, batteries and nodes. We then introduce each of the three entities separately. Batteries and cables are modelled using a parameter-based approach, while nodes are modelled using real-world data. This chapter serves as the introduction into technical aspects that are relevant later on, and also introduces the notation that is used throughout this report.

3.1 Storage-Cable-Node model

Here we introduce the **Storage-Cable-Node model** (Figure 3.1). This basic model consists of a battery, a cable and a node, each of which are discussed later in this chapter. In this section we explain the workings of this model, and introduce relevant terms. An effort was made to name these terms in accordance with their default notation in physics (see Table 3.1 for an overview of all terms).

The Storage-Cable-Node model consists of 3 main entities: the storage unit or battery B, the electricity cable c and a node W. W represents a source/sink of energy, and may model a neighbourhood, household, solar installation, factory or something else. W is connected to the main electricity grid through cable c. As W can represent different entities, c logically also can be a power cable located in the distribution (low- and

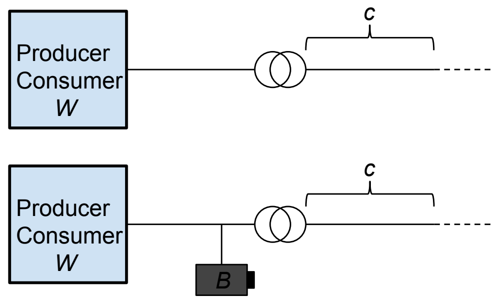


Figure 3.1: The Storage-Cable-Node model, consisting of a production/consumption unit W, power transformer and cable system c, and storage unit B (bottom). To allow comparisons, in some cases no storage unit B is present (top).

medium-voltage) or transportation (grid). c even doesn't have to be an actual cable, as it could represent a different bottleneck entity in the grid as well. For instance, expensive electricity transformers could also be damaged from handling high power. Lastly, the model may include a storage unit B. We will assume in this chapter B is a battery, but B can easily represent some other way of storage as well. In this model, W has the potential of damaging c by demanding or supplying high electricity rates.

Although electricity, demand, charge rates and other variables relevant to the model are analogue units, this study uses a discretion of time into 15-minute intervals. We chose 15-minutes as interval length for three reasons. First, 15-minutes represents the shortest time-interval in any European electricity market. In the Netherlands, 15-minute windows are used on the TenneT Imbalance market. Secondly, 15-minute intervals are commonly used in registering load demand data. Two major data sources in this study contain electrical load data for 15-minute intervals. Lastly, with the introduction of smart-meter data, 15-minute intervals have been established by law as the maximum query resolution on households. This means that in the coming years, 15-minute intervals will be widely used to analyse load demand and new energy innovations.

For each interval t, P_t^W is the demand or supply of power from W, where a positive value indicate a power consumption, and a negative value indicates production of electricity. The battery B also has a demand/supply of power at t, denoted by P_t^B. If P_t^B is positive the battery consumes (charges) electricity, if P_t^B the battery supplies (discharges) electricity. The cable c sustains the sum of powers that go to B and W such that the electricity flow is balanced:

$$P_t^W + P_t^B + P_t^c = 0, \qquad \forall t \tag{3.1}$$

In case no battery is present, we have P_t^B=0 such that

$$P_t^W = -P_t^c, \qquad \forall t \tag{3.2}$$

The Storage-Cable-Node model is a simplification of electricity grids which are much more complex networks in the real-world. However, in Section 4.2.3 we introduce earlier studies which have utilised the same types of model networks to explore the value of storage units in electricity grids. In Section 8.2 we briefly discuss incorporating more complex electricity grids.

Table 3.1 provides an overview of all the parameters introduced in this chapter. Below, we will separately discuss the three components of the Storage-Cable-Node model, battery B, cable c and node W.

P	Power.
C	Capacity.
t	Time interval t specifies a discrete window of 15 minutes.
T	When a finite time-horizon is used, T denotes the number of intervals t.
	Demand and Supply
P_t^W	The supply or demand rate from unit W at time t (kWh).
	Battery properties
P_t^B	The supply, or discharge rate, from the battery B at time t (kWh).
C_t^B	The capacity (amount of stored energy) of battery B at time t (kWh).
$C_{\max}^B$	Maximum capacity that can be stored in battery B (kWh).
$P_{\max}^B$	The maximum charge rate of battery B (kW). For all batteries, $P_{\max}^B$ will be positive, indicating its maximum discharge rate.
$P_{\min}^B$	The minimum charge rate of battery B (kW). As P_t^B indicates the supply by the battery, the minimum value will be negative, and specify the maximum charge rate of the battery.
$\mathcal{E}$	Round-trip efficiency of a battery, i.e. the fraction of electricity discharged per electricity charged.
	Cable usage
P_t^c	The supply rate of the connected power grid (G) at time t (kW).
$P_{\max}^c$	Maximum capacity (power) of cable c (kW).

Table 3.1: Overview of terms in the Storage-Cable-Node model.

3.2 Batteries

Batteries, or any other electricity storage devices are capable of storing electricity before discharging the electricity at a later time. There is an ongoing development in the battery industry where many materials and technologies are used to create batteries with high storage capacities, efficiency rates and lower costs. As this study focuses on the computational aspect of using storage units, and not the storage units itself, we utilise a relatively simple model for batteries.

Each battery B has a maximum storage capacity $C_{\max}^B$. The battery charge at the end of interval t is denoted by C_t^B. Both $C_{\max}^B$ and C_t^B are specified in kWh, the unit of electrical energy. At the end of each interval t, the battery charge level is within the range $[0, C_{\max}^B]$:

$$0 \leq C_t^B \leq C_{\max}^B, \qquad \forall t \tag{3.3}$$

A battery also has a limited range for the charge/discharge rate P_t^B (kW). The maximum discharge rate and maximum charge rates are indicated by $P_{\min}^B$ and $P_{\max}^B$ respectively such that:

$$P_{\min}^B \leq P_t^B \leq P_{\max}^B, \qquad \forall t \tag{3.4}$$

As $P_{\min}^B$ specifies the maximum discharge rate, this value must be negative. Otherwise, the battery is not be capable of discharging electricity. The battery also has a round-trip efficiency $\mathcal{E}$. The round-trip efficiency $\mathcal{E}$ specifies the amount of electricity that can be discharged relative to the original amount of electricity charged. For instance, charging a $\mathcal{E} = 0.75$ battery for 100 kWh only allows 75 kWh to be discharged, as 25 kWh is lost due to the inefficiency of the battery. For convenience we always calculate this loss of electricity when charging the battery, such that the battery charge C_t^B indicates the actual amount of electricity that still can be discharged:

$$C_{t+1}^B = \begin{cases} C_t^B + \frac{1}{4}\mathcal{E}P_t^B & P_t^B > 0 \\ C_t^B + \frac{1}{4}P_t^B & \text{otherwise} \end{cases} \tag{3.5}$$

Batteries are known to lose some of their charging capacity $C_{\max}^B$ over time. Furthermore, the use of the battery (when and how much the battery is charged/discharged) influence the long-term charging capacity and efficiency of the battery. These processes depend on the types of materials used, types of power inverters used as well as external factors such as temperature and humidity of the environment. In this study we chose not to model these complicating factors for three reasons. First, we are not experienced in the field of battery management, and inspection of literature available shows that there is a plethora of techniques and theories available which require experience to be judged accordingly. Second, many of these complications are battery-type specific and do not lend themselves well for a general battery model. Third, this study focuses on the added value of storage in electricity grids. We therefore focus on the return on a battery investment. The height of investment can subsequently indicate whether or not a battery is economically favourable. Decision makers together with battery experts can therefore use this study to see if the cost of a battery in a specific situation is preferred.

3.3 Cables

In the Storage-Cable-Node model, electricity cable c connects the node and battery to the external electricity grid. Running electricity through a cable causes the cable to heat up. In physics, this phenomena is called *Joule heating*. If the temperature of the cable becomes to high, some of the materials become damaged. For instance, the synthetics that isolate the cable core can melt. Therefore, typically every electricity cable has a specified maximum capacity $P_{\max}^c$ (kW), which specifies how much power the cable can sustain without causing damage to the cable. This maximum capacity is determined by the type of material the cable is made from and the thickness of the cable. If the cable is never used for more than the maximum capacity $P_{\max}^c$, the cable material will degrade in such a fashion that the cable can be used for the entirety of its life expectancy. For regular cables, the expected lifetime at normal use lies around 50 years.

There exists extensive literature on modelling how this excess heat shortens the lifetime of the cable. For instance, exceeding the cable maximum for short bursts of time tends to cause damage less compared to continuous over-use, as the cable is able to cool down in between [5]. Many other factors, such as the material used, the depth of the cable, surrounding materials and heat disposal also influence the heating and cooling down of cables as a result of electrical use. In this thesis we attempted to include realistic behaviour in our simulations, while using a simple model for cable damage, the doesn't go into detail about such conditions. Below, we will introduce two models that relate cable (over-) usage to the increased degradation of cables.

3.3.1 Joule heating and the quadratic damage model

Joule heating (or Joule's first law) "*describes the process where the energy of an electric current is converted into heat as it flows through a resistance.*" [25]. This is the main source for heat that affects the temperature of an electricity cable. Joule heating is described by:

$$H \propto I^2 \cdot R \cdot t \tag{3.6}$$

where H is the amount of heat produced, I the electrical current, R the resistance of the cable and t the time [26]. For a fixed interval, the added heat thus depends on the cable (resistance R) and the use (power P). Although the resistance of the cable R is not a constant, here we assume it is. According to van Oirsouw and Cobben [6], the dominant effect on the cable resistance is caused by temperature. In extreme cases, the resistance of the cable can change 10% between cold and hot conditions. We chose to ignore this effect to keep the model simple and easy to use. More complex calculations are performed by grid operators to fit the dimension of grids, but for the storage-based grid protection discussed in the next chapter, the complexity of this model is sufficient.

Since we now have resistance R, for a fixed time interval we have.

$$H \propto I^2 \tag{3.7}$$

In electricity grids the voltage is kept constant, so as $P = UI$ (Ohm's law) [27], we have

$$H \propto P^2 \tag{3.8}$$

Note that this law should not be confused with $P = I^2R$. This law would dictate a linear relation between H and P. However, $P = I^2R$ specifies the power of *loss*, and not the electrical power used in the cable.

If we assume that heat and cable damage have a linear relationship, we can use rule of Joule heating to formulate a power to damage relation $D(P)$:

$$D(P) = \begin{cases} \frac{1}{4}(P - P^c_{\max})^2 & P > P^c_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Here, $D(P)$ is the damage measure, P is the (absolute) power running through cable c in kW and $P^c_{\max}$ is the maximum capacity of cable c in kW. When the power in c exceeds the maximum $P^c_{\max}$, $D(P)$ increases quadratically compared to the difference in power between P and the maximum. We therefore call this model the quadratic damage model. As a result of this formula the unit for harmful energy is $(kW)^2h$.

Equation (3.11) specifies that only the power that exceeds the cable maximum counts towards cable damage. However, all power that runs through the cable contributes towards the heat. But as cables are designed to sustain a certain temperature (which is reflected in the maximum capacity $P^c_{\max}$, we ignore all electricity that is run below this maximum. Hence, $D(P)$ only focuses on the excess power, which would lead to excess temperature, which leads to cable damage.

The assumption that there is a linear relation between temperature and cable damage is made as there is no consensus on how cable damage can be quantified over a complete cable. Naturally, not all parts of the cable heat up equally, and not all parts of the cable can sustain high temperatures as well as other parts. Some authors have noted that currently there is no consensus about a standard way to quantify cable damage from overheating a cable.

3.3.2 Temperature modelling

In Höning et al. [5] it was shown that not only the amount of energy that is run through a cable, but also the pattern in which this occurs greatly affects the cable temperature. In short, the authors showed that exposing the cable to short bursts of electricity resulted in lower cable temperatures. The quadratic model provides a straightforward cable damage calculation, but lacks the behaviour described in Höning et al. [5]. The damage model can be extended further to model this behaviour, by incorporating temperature directly.

We extend the model by keeping track of the temperature of cable c at interval t, T^c_t. Use of the cable above the maximum capacity $P^c_{\max}$ (kW) will increase the temperature. Coincidentally, running zero or low-level power through the cable causes a heated cable to subsequently cool down. We therefore define model that keeps track of a temperature T^c_t for cable c at time t:

$$T^c_{t+1}(P) = \begin{cases} T^c_0 + \phi(T^c_t - T^c_0) + \varphi(P - P^c_{\max})^2 & P > P^c_{\max} \\ T^c_0 + \phi(T^c_t - T^c_0) & \text{otherwise} \end{cases} \tag{3.10}$$

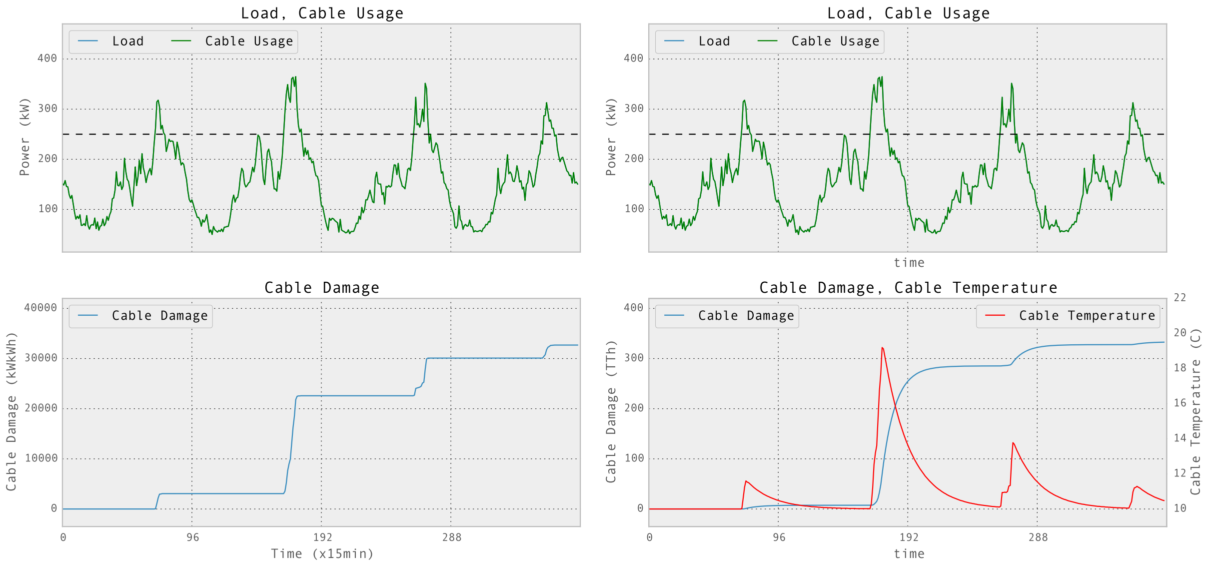


Figure 3.2: Comparison between the behaviour of the quadratic and Temperature model for cable damage. Exceeding the cable maximum power capacity $P^c_{\max}$ results in cable damage. In the Quadratic model, this translates directly to cable damage. In the Temperature model, the state (temperature) changes, causing the cable damage.

In this model, the temperature of cable c at new time interval $t+1$, T^c_{t+1}, is a result of the sum of (1) the original starting temperature T^c_0, (2) the temperature at the last interval, cooled down at the fraction ϕ and (3) added temperature as a result of exceeding the cable maximum capacity $P^c_{\max}$. The model explicitly specifies three parameters. The first, T^c_0 is the base temperature of the cable's surroundings. At the first interval ($t = 0$), the cable has this temperature. If the cable is not used, the cable temperature will always return to T^c_0 eventually. The second parameter is the cool-down coefficient ϕ. Each interval, the *added* temperature above T^c_0 becomes the fraction ϕ of the previous time interval. The third parameter specifies the rate in which the cable heats up as a result of exceeding the cable maximum capacity. This parameter, the heat-coefficient φ converts the excessive power run through c into added heat. Note that this formulation incorporates the quadratic behaviour dictated by Joule's first law.

The above model provides a measure for determining the temperature of a cable throughout our calculations. However, linking this temperature to a damage measure requires an additional step. The cable damage in the temperature model can be calculated as follows:

$$D^t = \begin{cases} \frac{1}{4}(T^c_t)^2 & T^c_t > T^c_0 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

The cable damage inflicted on cable c at time t is a result of sustaining temperature T^c_t. As a result of this quadratic formula, the unit of D^c_t in the temperature model is (TTh).

Figure 3.3 demonstrates how this new damage model reacts differently compared to the quadratic damage model. The temperature model calculates less cable damage when the excess power is spread over time compared to long stretches of continuous overheating. This model is in line with earlier found behaviour in Höning et al. [5].

The temperature model contains parameters that influence the damage calculation. These parameters relate to the type of cable, the environment, how deep or shallow the cable is buried in the ground and many other

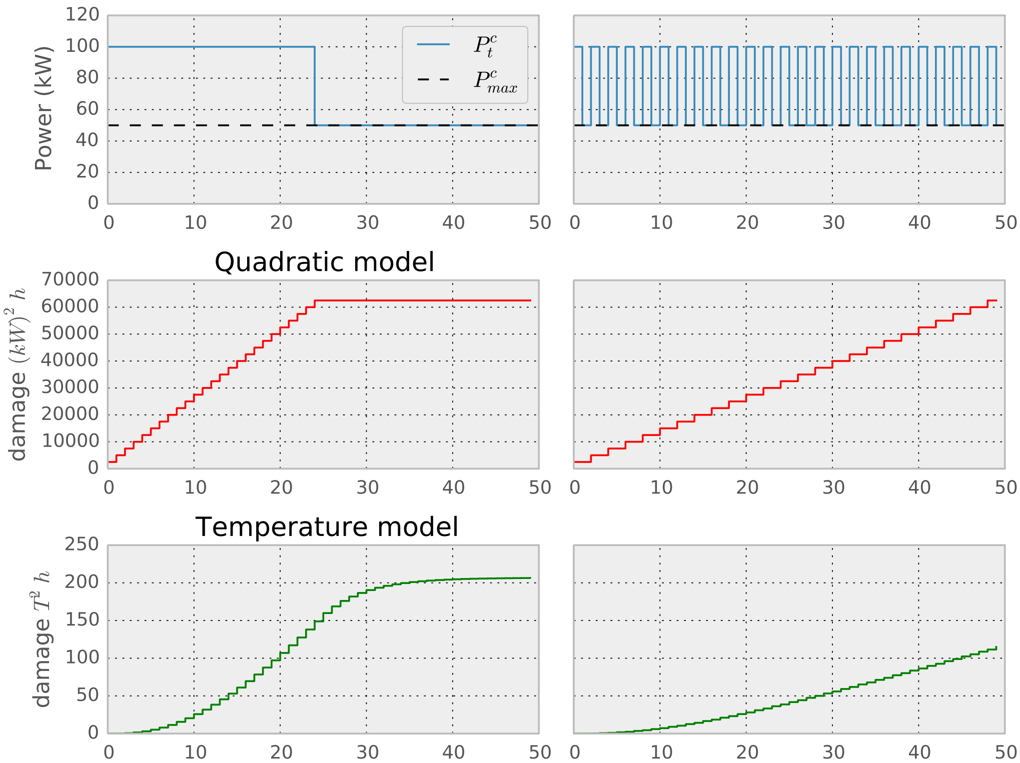


Figure 3.3: Comparison how the cable usage pattern influences the cable damage in the quadratic and temperature damage models. The top row specifies the cable usage. Both scenarios contain 25 intervals of 100 kW demand, and 25 intervals of 50 kW demand. In the quadratic damage model (middle row), the final damage caused is equal in both cases. In the temperature model the cumulative damage is almost two times as small in the latter pattern.

factors. Although we would like to fit these parameters using real-world experiments or literature studies, our resources do not allow such a modelling in this study. However, we believe that using the temperature model and its described behaviour allows use to more realistically evaluate the benefits of energy storage, even if we use estimated parameters.

3.4 Nodes

While batteries and cables are modelled using parameter-based models to explain their behaviour, our approach to modelling nodes in completely data-driven. Nodes simply are connected to the electrical network and either produce or consume some amount of electricity at one or more points in time. This allows for a uniform approach in simulations and calculations, as the objective always remains to supply the node with enough energy (or to consume its produced energy) through the grid without causing any damages. The demanded or produced energy is often referred to as the **load** of a node. Prediction on the future load of a node is referred to as **load forecasting**.

In order to include realistic entities (households, neighbourhoods, companies, industries, etc.) in this study, we rely on external data sources. In Sections 3.4.1 and 3.4.2 we therefore introduce data sets that model electricity consumers and photovoltaic producers, respectively. We also provide brief analysis and discuss which real-world entities could be represented using the data. Section D.1 provides a table listing all data sources used in this study.

3.4.1 Electrical Load Demand Consumers

To model the load demand (or energy consumption) by households, companies, utilities or other entities, we obtained three independent data sets. Below we introduce the *Zonnedael* data set, which contains electricity demand data from 75 households. The *Schiedam* data set contains electricity use data from 69 municipal utilities, which can be used for modelling non-residential, non-industrial energy use such as office buildings, utilities or sporting venues. The Energy Data Services Netherlands (EDSN) data set is an official prognosis of the electricity use for different kinds of users, most of which are households. Below, the three data sources are briefly introduced and analysed.

Zonnedael

The *Zonnedael* data set was obtained from the DSO Liander website and contains the data from 80 Dutch households [28]. For every 15-minute interval in 2013 Liander registered the total load of each of these households using smart meter (Dutch: *slimme meter*) systems. The households were located in different locations in the Netherlands, although the locations were removed for privacy. For each of the households the home type and home age are known. 5 of the households in the data set contain missing values, therefore these are left out in the analysis in this study. Some of the houses in the data set include solar installations, resulting in energy production at certain times.

As expected, the data shows a both a daily (Figure 3.4A) and weekly (Figure 3.5) pattern as energy consumption highly correlates with the daily and weekly patterns of people's lifes. Note that in both these figures the energy consumption was summed over all 75 households. (Figure 3.4A) also shows that at night (from 22:00-06:00), the daily load is much more constant compared to during the day, as the relative point distribution is much more defined in this time window. The patterns in Figure 3.5 highlight differences in energy consumption between the days of the week. The main difference in load demand is the consumption during daytime between weekdays and the weekend. From Friday until Sunday there is a steady rise in energy consumption between 12:00 and 16:00. We speculate this has to due with most residents being home, in contrast to being at work during the week on these days.

If we zoom in individual households (Figure 3.4B) we see a wide range in yearly energy consumption, ranging from -2000 kWh to $> 10,000$ kWh. These negative totals could be caused by a combination of a minimal energy use and a large solar installation. Figure 3.4B shows that apart from these outliers, households typically consume between 2000 kWh and 4000 kWh on a yearly basis.

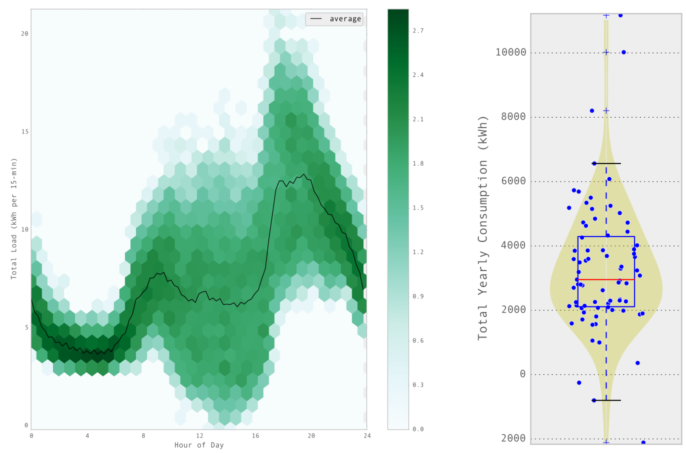


Figure 3.4: **(A)** Hexagonal Bining Plot of Cumulative Residential Load ($n = 75$) in the *Zonnedael* data set. Hexagons are colored using a log-scale. **(B)** Violin Plot of Total Yearly Energy Consumption per Household in the *Zonnedael* data set.

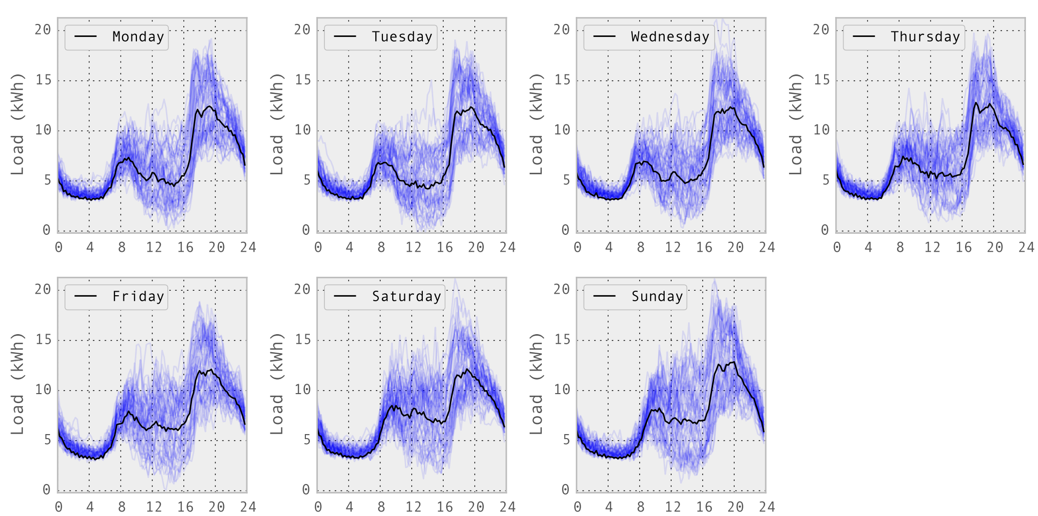


Figure 3.5: Cumulative Residential Load ($n = 75$) in the *Zonnedael* data set, per day of the week. Blue lines reflect individual weeks, the black line marks the average load.

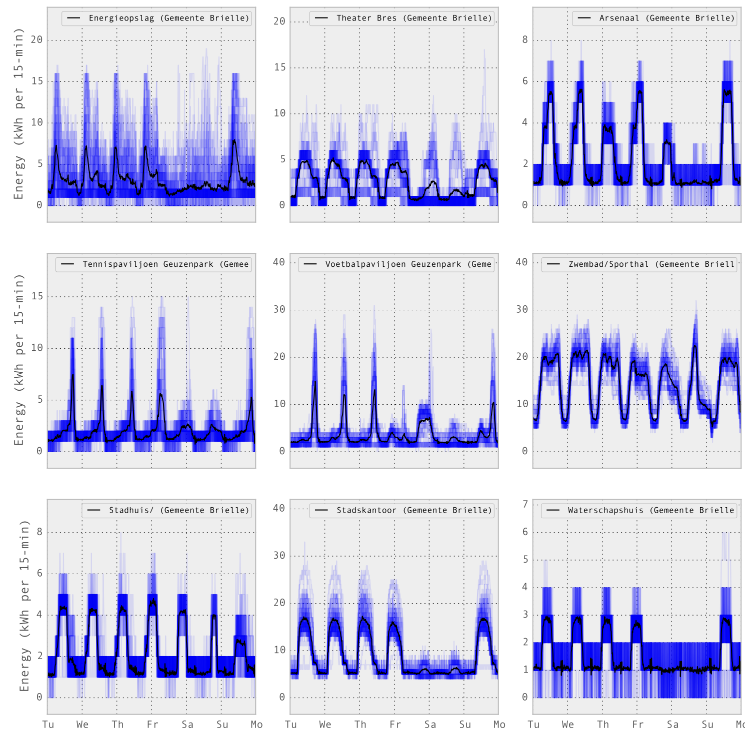


Figure 3.6: Load Demand Characterisation of 9 out of 69 Utilities in the *Schiedam* data set. Featured from top to bottom, left to right: Energy Storage, Theatre, Library (*Arsenaal*), Tennis Club, Soccer Club, Swimming Pool, City Hall, City Administration, Water Management Office. Blue lines reflect individual weeks, the black line marks the average load.

Schiedam Government Buildings

The *Schiedam* data set was obtained from the TenderNed website [29]. The data was taken from an application by the municipalities of Schiedam, Brielle and others[1] to collectively purchase electricity. For this application, they provided 15-minute interval data of 69 municipal utilities from 2013. Figure 3.6 shows typical patterns for some of these utilities. By itself, this data often tells a story about the nature of the utility. For example, we can see that the library *Arsenaal* is closed on Sundays, while city hall is still used. The tennis and soccer clubs uses electricity mainly at night (likely for lights), but not so much on weekends, where the daytime energy consumption is higher. Here we see that energy consumption is higher at times people are more likely to enjoy their free time. This data can be included in our study to diversify the energy patterns, as opposed to residential load data. In practice, not all entities in the electricity system will demand power at the same time. This data allows the modelling of government buildings and installations, but also general office buildings and small companies.

EDSN Profiles

EDSN is a company that analyses the use of electricity in the Netherlands. EDSN is responsible for forecasting the standard profiles of energy use for small to mid-sized connections (households, companies, office buildings). Each year these profiles are produced, and used by the TSO, DSOs and electricity- and gas-companies to provide sufficient electricity and gas to the consumers. These profiles have a 15-minute accuracy and

[1]Schiedam, Brielle, Hellevoetsluis, Maassluis, Spijkenisse, Vlaardingen and Westvoorne [29].

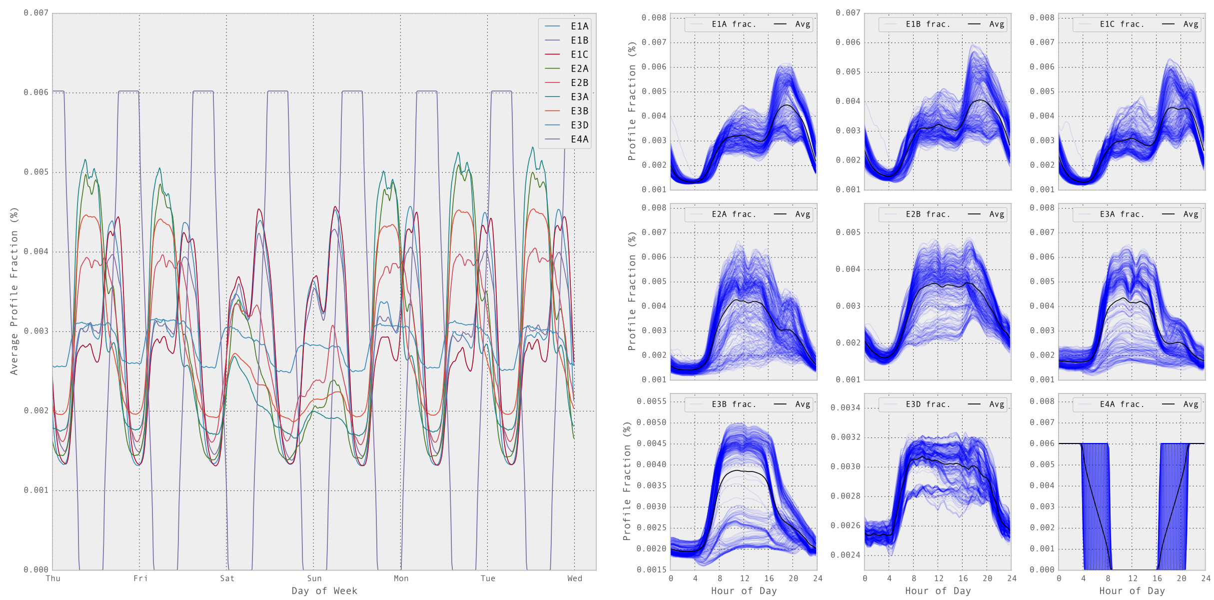


Figure 3.7: EDSN profiles for 2015. **(A)** Average Weekly Fraction Curves for the EDSN categories. **(B)** Daily Fraction Curves for each of the 9 EDSN categories. Blue lines reflect individual weeks, the black line marks the average load.

reflect monthly, daily and seasonal behaviour. Although this data is a projection, we can use the profiles to manufacture artificial electricity use by adding noise and additional data on top of these profiles.

9 publicly available EDSN profiles for 2012-2015 were used in this study. The patterns in this data are visualised in Figure 3.7. The profile codes stand for different size and type of users, see [30] for detailed descriptions. Category E4A is the estimated electricity profile of street lights.

3.4.2 Photo Voltaic (PV) Electricity Production

To research the extend in which energy storage can facilitate or balance the introduction of Photo Voltaic (PV) in electricity grids, we model nodes with real-world data. PV Installations can typically be characterised by a limited number of properties. Of these properties, the **peak power** (kWp) indicates the amount of energy an installation produces under full solar radiation (defined as 1000 watts per square meter [31]). This study makes use of two separate sources of PV production data. Unfortunately, we were not able to obtain 15-minute interval data for a sufficient time period to match with load data presented in Section 3.4.1. However, the two obtained data sources allowed for producing relevant simulation data, as explained in Chapter 6.

Solar Log

The Solar Log is an online initiative where owners of PV installations share the energy production of respective installations. Although the website does not allow for a straightforward download of the data, we were able to scrape the data using a script (see Appendix E.2). This Python script was extended from a version included in [32]. Using the script, we were able to collect the properties of over 250 separate installations. Furthermore, for each of these installations we stored the daily energy production (both in alternating current (AC) and direct current (DC)) for all present values over the past 5 years. The website did not allow for accessing information at smaller intervals.

Moraitis

The Moraitis data set contains 5-minute interval data from December 27th 2013 until October 22nd 2014 and was obtained by a request to a fellow Utrecht University student P. Moraitis. The data set represents over 45,000 data points. The data set gives us a typical example pattern for the generation of solar electricity, with a strong influence of local effects. The *CentroSolar* installation was installed in 2012. It has a 7.68 kWp capacity. The installation is located at (longitude, latitude) (5.734, 51.197), and situated 2.100m away from the nearest KNMI station 377.

Chapter 4

Energy Storage as Grid Enhancement

Contents

As introduced earlier, electricity storage has the potential to prevent electricity grid damage. In this chapter, we will introduce the concept of grid enhancement through electricity storage. In short, we show how we can prevent cable damage using a battery, and remove the need for grid enforcements. In addition, we show how we can quantify the economical benefit of storage in electricity grids.

4.1 Demonstration

In electricity grids, cable damage can occur when the sum of electricity demanded or supplied from node W exceeds the cable maximum (see Section 3.3). The quadratic cable damage model from Section 3.3.1 specifies that the damage D is a function of the electricity running though the cable P:

$$D(P) = \begin{cases} \frac{1}{4}(P - P^c_{\max})^2 & P > P^c_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$D(P)$ specifies the cable damage (in $(kW)^2h$) given an absolute power running through c, P, and cable maximum capacity $P^c_{\max}$. Consider the Storage-Cable-Node model from the previous chapter, with node W, battery B and cable c. We have T time intervals t ($1 \leq t \leq T$). The total damage to cable c during the intervals is

$$\sum_{t=1}^{T} D(|P^W_t + P_t|) \tag{4.2}$$

Here, P_t is the charge rate of battery B. By discharging during at which P^w_t exceeds $P^c_{\max}$, and by charging during intervals at which $-P^W_t < -P^c_{\max}$, battery B can prevent cable damage. The objective now becomes to minimise the damage to cable c over the T intervals by deciding on a **charge schedule**, a series of charge rates decisions $P_1^B, .., P_T^B$. Before we can formulate an algorithm to determine the optimal charge schedule, we must provide a more formal problem definition.

4.1.1 Problem definition

$$\text{minimise } K(P_1..P_T) = \sum_{t=1}^{T} D(|P_t^W + P_t|) \tag{4.3}$$

$$\text{s.t.} \tag{4.4}$$

$$P_{\min}^B \leq P_t \leq P_{\max}^B \qquad 1 \leq t \leq T \tag{4.5}$$

$$0 \leq C_0^B + \frac{1}{4}\sum_{k=1}^{t} P_t \leq C_{\max}^B \qquad 1 \leq t \leq T \tag{4.6}$$

$K(P_1...P_T)$ is the total cable damage that occurs over T intervals given the charge rates $P_1, .., P_T$ (for simplicity, P_t denotes P_t^B in the rest of this chapter). In addition to the terms defined above, $[P_{\min}^B, P_{\max}^B]$ is the interval of possible charge rates, C_0^B is the initial battery fill and $C_{\max}^B$ is the maximum battery capacity.

We can find an optimal charge schedule using dynamic programming by defining:

$$F_t^k = \text{minimal cable damage to reach interval } t \text{ at a battery charge } C_t^B = k \tag{4.7}$$

$$= \min_{\{P_1..P_t\}} K(\{P_1..P_t\}) \tag{4.8}$$

as the minimal cable damage necessary to reach interval t with battery charge $C_t^B = k$. Consider the recurrence relation:

$$F_0^k = \begin{cases} 0 & k = C_0^B \\ \infty & \text{otherwise} \end{cases} \tag{4.9}$$

$$F_{t+1}^k = \min_{j \in J_k} \left[F_t^{k-\frac{1}{4}P_t} + D(|P_t^W + P_t|) \right] \tag{4.10}$$

where

$$J_k = \{ P_t \mid P_{\min}^B \leq P_t \leq P_{\max}^B \wedge -4k \leq P_t \leq 4(C_{\max}^B - k) \} \tag{4.11}$$

Theorem 4.1.1. *F_t^k is the minimal cable damage necessary to reach interval t with a battery charge of k.*

Proof. Proof by cases:

- *Case* $t = 0$: At t=0, K() =0. The only requirement is that the battery charge equals k. Since the battery charge at $t = 0$ equals C_0^B, $F_0^k = 0$ if $k = C_0^b$. For other battery charges k, F_t^k is infeasible.
- *Case* $t + 1$: At t+1, the cable damage equals the cable damage during interval t+1, and the cable damage during earlier intervals. The minimal cable damage thus is the minimal sum of those two damages. For any charge rate P_t, the cable damage $D(P_t^W + P_t)$ is fixed. Additionally, $F_t^{k-\frac{1}{4}p}$ is the minimal damage necessary to reach the necessary battery charge, in order to be able to charge $P_t + 1$ at this interval. Therefore, the minimal cable damage necessary to reach interval $t + 1$ with battery charge k, is the minimal sum of $F_t^{k-\frac{1}{4}p} + D(P_t^W + P_t)$ over all possible charge rates P_t.

□

In order to find an optimal charge schedule, we only need to calculate the optimal values F_t^k for $0 \leq k \leq C_{\max}^B$, $1 \leq t \leq T$, and backtrack the charge rates P_t at each interval.

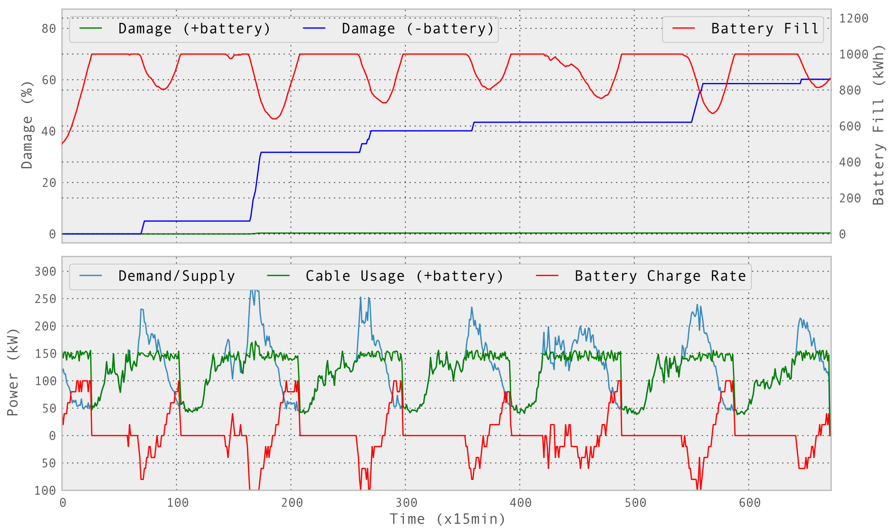


Figure 4.1: Cable damage test using Zonnedael data, week 0. *(top)* Battery fill (red) in kWh and cable damage in the situation with (green) and without (blue) a battery. *(bottom)* Demand/supply by W in kW, batter charge rate in kW and cable usage in kW. In the situation without a battery, the cable usage equals the demand/supply of W.

4.1.2 Results

We can now demonstrate how a battery can prevent cable damage. We will run a calculation on data from the Zonnedael data set (Section 3.4.1) to simulate the node W. Consider a cable with $P^c_{\max} = 150kW$, and a battery with $[P^B_{\min}, P^B_{\max}] = [-100, 100]$, $C^B_{\max} = 1000kWh$. For each week in the Zonnedael data set, we calculated the total cable damage with or without the battery. An example calculation can be seen in Figure 4.1[1]. The total cable damage per week is visualised in Figure 4.2.

Without a battery, the total damage is 902928 $(kW)^2h$. Figure 4.2 also shows that this damage mainly occurs during fall and winter, when the electricity use is apparently much higher. If we include a battery in the model, the cable damage reduces to 798 $(kW)^2h$, a $>$ 1000 fold decrease of cable damage.

These results can directly be used to calculate the economic benefit of installing batteries in electricity grids. For instance, in Section 3.3 we discussed how a cable can only sustain a certain amount of cable damage during its lifetime, depending on its material and thickness. If we take the $12.25 * 10^4$ $(kW)^2h$ maximum cable damage from the example calculation, we see that in this calculation, the cable would be completely destroyed. However, we note that the $P^c_{\max} = 150kW$ is unrealistically low and much higher compared to the maximum demand by W in practice. Still, using the calculation and the maximum cable damage, we can express the cable damage in a number of years of which the cable lifetime decreases. So if the cost of a cable is X per year, and the battery unit prevents cable damage such that the cable can be used for n more years, the battery unit is economically profitable if it can be implemented for less than nX.

We can say that implementing electricity storage allows W to demand/supply more electricity without damaging the cable c. This is what we call **Grid Enhancement**, which is seen as a possible solution to the increasing demand in electricity in domestic electricity grids and the tendency to consume electricity in shorter amount of time. Below, we will summarise the efforts that have been done in finding optimal charge

[1]See **http://tweedejan.synology.me/simulation/results/offline.html** for the other weekly plots.

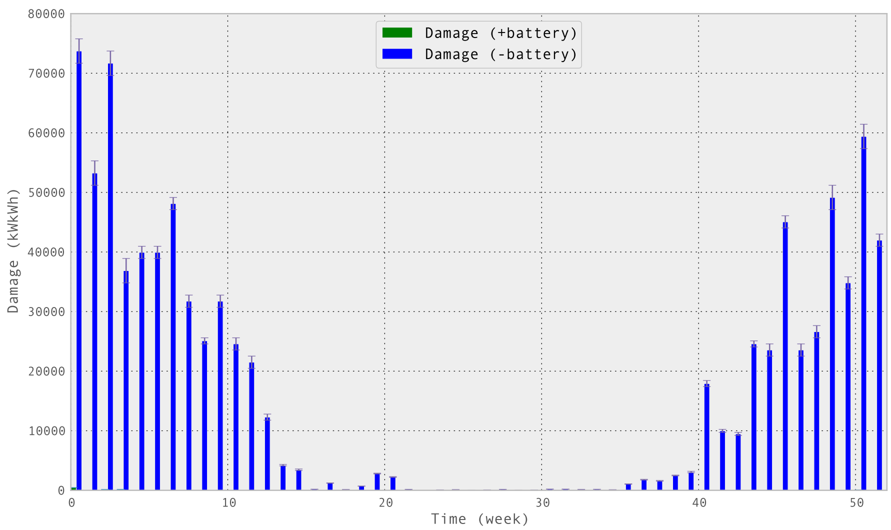


Figure 4.2: Weekly cable damages using the *Zonnedael* data simulating. As a result of implementing a battery unit (+battery), the cable damage drastically decreases. In total, the situation with battery inflicts 798 $(kW)^2h$ damage, while without a battery the damage sums to 902928 $(kW)^2h$ (a >1000 fold increase).

schedules in order to provide this grid enhancement. As a part of this problem, variable electricity prices are included. In the next chapter, we will focus on algorithms that optimise both grid enhancement as well as variable-priced electricity consumption.

4.2 Grid Enhancement as an Optimisation Problem

Now that we have introduced energy storage as grid enhancement, we can discuss past efforts in analysing the optimisation problems associated with this application. Reducing peak demands in electricity grids, peak-shaving is a well-described technique. Bar-Noy et al. [1] first approached peak-shaving as an optimisation problem. The objective in their formulation was to reduce the maximum electricity flow in the network. In terms of the model from our demonstration, the objective can be formulated as:

$$\min_{P_1..P_T} \left[\max_{t=1}^{T} \left(|P_t^W + P_t| \right) \right] \tag{4.12}$$

In other words, the objective is to lower the maximum demand on the cable, regardless of the maximum cable capacity. This objective is central to **peak-shaving**, it reduces the maximum value of a peak in electricity by increasing the electricity flow in other intervals. Electricity storage is able to reduce the cable demand on peak intervals, and to increase the cable demand on other intervals, and thus is capable of peak-shaving.

Peak-shaving using electricity storage as an optimisation problem is closely related to Economic Lot Sizing (ELS), a classic operations research problem. Before introducing different aspects of the grid enhancement problem, we will introduce ELS, including its relation to peak-shaving and grid enhancement.

4.2.1 Economic Lot Sizing (ELS)

The Economic Lot Sizing problem (ELS) entails a class of optimisation problems in Operation Research that involve production planning to satisfy demand patterns. The main optimisation problem in this class, *Single-Item Lot-Sizing*, is the planning of the amount in which a single product should be produced in a number discrete intervals. The produced amounts should satisfy a specified demand in each period. Although many varieties of the ELS problem class exist, the minimisation of the sum of different costs is ubiquitously the objective. These cost may include production, holding (keeping products in inventory) and set-up (initial cost for producing) costs.

A formal definition of ELS is provided below:

$$\min \sum_{t=1}^{T} (f_t y_t + p_t x_t + h_t I_t) \tag{4.13}$$

$$\text{s.t.} \tag{4.14}$$

$$x_t + I_{t-1} = d_t + I_t \tag{4.15}$$

$$I_0 = 0; I_T = 0 \tag{4.16}$$

$$x_t > 0 \rightarrow y_t = 1 \tag{4.17}$$

$$x_t \geq 0 \tag{4.18}$$

$$I_t \geq 0 \tag{4.19}$$

$$y_t \in \{0, 1\} \tag{4.20}$$

For a number of periods $t(1 \leq t \leq T)$ we have a demand d_t of products that are removed from the inventory. At each interval t, we can produce at p_t cost per unit. x_t is the amount of produced products at t. Producing any products in t requires a set-up cost of f_t. A decision variable $y_t \in \{0, 1\}$ indicates whether production occurs in t. The inventory after period t is denoted by I_t. The inventory in $t + 1$ is determined by the production x_t, past inventory I_t and demand d_t. In order to keep a single product in the inventory in t costs h_t. The amount of production x_t and inventory I_t in each period t are the decision variables in ELS [33].

Capacitated Lot Sizing Problem (CLSP) The above definition is a widely used as the general definition of the ELS [34, 33, 35, 36]. This model is often referred to as the *uncapacitated* ELS problem [37, 38, 33, 39]. In the *capacitated* version of ELS, there is a maximum production capacity c_t such that $0 \leq x_t \leq c_t$ for each $t = 1, ...T$. This problem is the Capacitated Lot Sizing Problem (CLSP) [38].

ELS with Bounded Inventory A further extension of the ELS is the class of problems where the total inventory I_t is bounded by a fixed number S. Although CLSP and the ELS with bounded inventory are mathematically related, the principles that are found in optimal plans for both problems are different [40]. The initial approach to the intentory-bounded ELS was published in Love [41], and further improved by [40] to a $\mathcal{O}(T^3)$ running time using dynamic programming. Both Love [41] and Gutiérrez et al. [40] use a concave function $p_t(x_t)$ in their models, as opposed to the unit-constant p_t described above.

It should be noted, that in relation to the CLSP, the bounded-inventory ELS has received far less academic attention. This might be due that in the classic ELS, inventory (holding) costs are already modelled, and that in only a few cases inventory is truly limited. Instead of using an inventory bound the holding costs could reflect for instance renting extra storage space.

In a few publications, the combination of both capacitated production capacities and bounded inventory is discussed. Sandbothe and Thompson [39] solve this problem in polynomial time using dynamic programming. However, their approach relies heavily on the assumption that the unit production costs p_t are defined by a concave function. Liu et al. [42] improve on this result, maintaining the aforementioned assumption.

Grid Enhancement and Economic Lot Sizing Consider the inventory in the ELS problem as energy storage. Producing products at interval t is analogues to charging a battery. In both cases, there is a cost involved (electricity price), a holding cost (possible efficiency loss) and maximum inventory level (charge

Objective	Used in	Description
$\min\limits_{P_1..P_T} \max\limits_{t=1}^{T} \left(\lvert P_t^W + P_t \rvert\right)$	Bar-Noy et al. [1], Nykamp et al. [4]	Classical peak-shaving.
$\min\limits_{P_1..P_T} \sum\limits_{t=1}^{T} D(\lvert P_t^W + P_t \rvert)$	Section 4.1	Minimise cable damage.
$\min\limits_{P_1..P_T} \frac{1}{4} \sum\limits_{t=1}^{T} P_t f_t$	Nykamp et al. [4]	Minimise electricity cost.
$\min\limits_{P_1..P_T} \omega \sum\limits_{t=1}^{T} v_t + \frac{1}{4} \sum\limits_{t=1}^{T} P_t f_t$	Ramezani et al. [3]	Minimise cable damage and electricity cost.

Table 4.1: Comparison of objective functions for peak-shaving and grid enhancement.

capacity) and production capacity (maximum charge speed). Of course, a battery also discharges, distinguishing gird enhancement from ELS. In Bar-Noy et al. [1], the authors note that peak-shaving is equivalent to capacitated inventory-bounded ELS where the objective would be not to to minimise costs, but to minimise the maximum production capacity.

In Economic Lot Sizing, production costs p_t may vary between intervals. For instance, the production cost could be dependent on a variable price of resources, or the availability of personnel. In Grid Enhancement, a related variable production cost occurs when electricity prices vary between intervals. Several publications that discuss energy storage as grid enhancement have stated that the economic profit of implementing storage could increase when the electricity is bought and sold using variable prices. In addition of only balancing the energy flow in a grid, the battery could buy electricity (charge) at a low price, and sell (discharge) the electricity at a higher price. If the difference in electricity price is larger than the loss of electricity due to battery efficiency, this could provide an additional profit for energy storage as grid enhancement [2].

4.2.2 Variable Electricity Pricing

Variable electricity prices create a potential extra profit to energy storage. However, maximising the revenue using price differences might be conflicting with the original grid enhancement objectives. Nykamp et al. [4] and Ramezani et al. [3] first discuss the optimisation problem which includes both grid enhancement and variable pricing. Nykamp et al. [4] compares these modes of storage usage separately. One algorithm tries to minimise the maximum cable usage (4.12), while their second approach minimises the total cost of energy:

$$\min_{P_1..P_T} \frac{1}{4} \sum_{t=1}^{T} P_t f_t \tag{4.21}$$

When using this objective, the cable maximum capacity is taken as a constraint for possible solutions. A battery schedule is only feasible if the cable usage is less than or equal to the maximum cable capacity in each interval. Based on real-world electricity prices and load demand, the authors show that there is a large difference in charge schedules when operating using these two approaches. In other words, the two objectives are conflicting. This indicates that electricity prices are relatively high during peak usage moments, when the demand is highest in residential areas. Operating the battery using the classic peak-shaving objective (4.12) will result in high electricity cost, making storage as grid enhancement less economically beneficial.

Using (4.21) as an objective, a potential problem arises when there is no feasible charge schedule. For instance, if a demand peak $P_t^w > P_{\max}^c - P_{\min}^B$, the cable maximum capacity will be exceeded regardless of the charge schedule. In such a case, cable damage occurs. Ramezani et al. [3] resolves this problem by combining the revenue-oriented objective with a cable damage component:

$$\min_{P_1..P_T} \omega \sum_{t=1}^{T} v_t + \frac{1}{4} \sum_{t=1}^{T} P_t f_t \tag{4.22}$$

Here, ω is a scaling-factor that equates cable damage to a monetary cost. v_t denotes the cable damage occurring in interval t. Objective (4.22) allows an optimisation of both cable damage and revenue. ω reflects the value of cable damage. Increasing ω puts emphasis on cable damage, and decreases the value of revenue. If cable damage is the only objective, a high value of ω can be used that neglects the relatively small revenue cost.

Although Bar-Noy et al. [1], Nykamp et al. [4] and Ramezani et al. [3] all discuss the value and benefits of energy storage to reduce cable damage, each uses a different objective function (Table 4.1). As noted by Nykamp et al. [2], the current cost of energy storage is relatively high such that storage of PV is currently only "*extremely positive conditions*". Therefore, we believe that the usage of variable pricing to lower the cost of storage will be a necessity for storage implementation in electricity grids. In the remaineder of this study we will therefore focus on minimising the cost of electricity. Since storage will be used to prevent cable damage, cable damage should always be prevented if possible. In terms of objective 4.22, ω should be high. In some instances, we will discuss the optimisation of revenue with preventing cable damage as a hard constraint. In later instances, we will include cable damage in the objective function, similar to (4.22).

4.2.3 Evaluating the cost of grid enhancement

In the previous section we introduced different objectives, all concerning grid enhancement using energy storage. Three of those publications ([4, 3, 2]) have provided methods for constructing charge schedules and subsequently evaluated the cost (both in cable damage and revenue) of those schedules.

Online/Offline evaluations In the introduction of the classic peak-shaving optimisation problem, Bar-Noy et al. [1] already distinguished algorithms for online and offline calculations. In computer science, an online algorithm processes input piece-by-piece, while offline calculations are performed with full knowledge of the problem. Both of these types of calculations can be used to evaluate charge schedules. For instance, Ramezani et al. [3] uses offline calculations to evaluate the performance of the online algorithm. The offline algorithm is a linear programming formulation that optimises the objective function (4.22). The online calculation is performed using two heuristics. The H_2 heuristic in this study is found to reach 83% of the theoretical optimum in their calculations. Therefore, we included the H_2 heuristic in this study. A Java implementation of this algorithm is provided in Appendix E.1. Both Nykamp et al. [4] and Nykamp et al. [2] evaluate the possible revenue of storage using offline calculations.

In practice, battery schedules should be made regularly. However, the exact load demand and electricity prices are not always known. We therefore argue that in order to evaluate the profitability of storage, algorithms need to be run in online evaluations. The continuous scheduling and real-time execution of charging dictate the real profit of storage. Such online evaluations require forecasting of load demand and electricity prices. In Ramezani et al. [3] these forecasts were produced using a simple average daily profile of electricity demand and price. However, the quality of these forecasts will influence the benefits of making charge schedules. Intuitively, when only bad-quality forecasts are available, scheduling based on these forecast will be of little use.

Models for Cable Damage Both Nykamp et al. [4] and Nykamp et al. [2] use the maximum cable capacity as a hard constraint on their charge schedules. This implies that in their calculations, a feasible charge schedule is always present. As explained in Section 4.2.2, this may not always be the case. Ramezani et al. [3] explicitly discusses cable damage as a result of exceeding the cable maximum capacity. Although the authors state that "*There is currently no consensus among engineering experts about a standard cost*

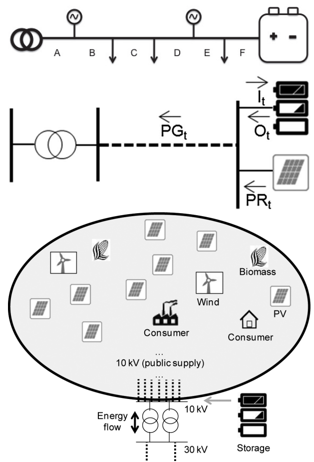


Figure 4.3: Comparison for grid models in (top) Ramezani et al. [3], (middle) Nykamp et al. [2], and (bottom) Nykamp et al. [4]

function that represents economic losses experienced by overheating a cable", they do propose a cost function depending on the power demand on cables. The authors propose a cost function

$$v(x,k) = \begin{cases} c_h^k & x > C \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

where x is the power running through the cable, k indicates the k-th interval where x exceeds the maximum cable capacity C and c_h the cost of doing cable damage. The authors chose a cost function for cable damage that focuses on increasing costs if the cable capacity is exceeded for consecutive intervals. Although there indeed is no simple, universally agreed-upon cost function for cable damage, the notion that cable damage in consecutive intervals is more harmful than short interspersed peaks has been shown.[5]

Calculating Costs and Benefits Finally, to evaluate the potential economic benefit of storage, the specific battery and cable properties need to be specified. Both Ramezani et al. [3] and Nykamp et al. [4] formulated situations where a battery was used to prevent cable damage and optimise revenue. It must be noted, that both publications, as well as the model used in Nykamp et al. [2] use a similar grid model (Figure 4.3). This grid model, introduced in Section 3.1 as the Cable-Node-Battery model, features a production/consumption site of energy (node), a storage device (battery) and a grid asset to protect. This grid asset could be a power cable, a power transformer or a different grid asset.

Details of the situations in which Ramezani et al. [3] and Nykamp et al. [4] evaluate charge schedules are summarised in Table 4.2. In both cases, the situation features a residential area. In Nykamp et al. [4], this area is enforced with significant PV production. The data comes from a German residential area. The electricity prices were also taken from two German electricity markets, the EPEX hourly intra-day and day-ahead markets. Although both markets were included, calculations were run on each market separately, showing that intra-day prices could potentially be more profitable due to larger price differences. Ramezani et al. [3] also features a residential area where a power cable asset is protected. The electricity prices were taken from the UK spot market.

Both publications use a 80% battery efficiency, which specifies that 80% of the charged energy can be discharged at a later time. This efficiency is realistic and has been achieved in multiple battery types [43, 6].

Interestingly, all publications regarding storage as grid enhancement do not consider demand-response solutions to take place as well. In some discussions, both demand-response and distributed storage have been presented to together solve grid issues. It is therefore surprising that studies focused on applicability

	symbol	Ramezani et al. [3]	Nykamp et al. [4]
Cable Capacity	$P^c_{\max}$	46 kW	6.43 MW
Battery Capacity	$C^B_{\max}$	315 kWh	8 MWh
Battery Charge Rates	$[P^B_{\min}, P^B_{\max}]$	[-5 kW, 5 kW]	[-2 MW, 2 MW]
Battery Efficiency	$\mathcal{E}$	0.80	0.80
Maximum Load Demand	$\max P^W_t$	100 kW	8.43 MW
Market Data		UK spot market	German EPEX intra-day, day-ahead market

Table 4.2: Comparison of the battery, cable and load demand specifications in Ramezani et al. [3] and Nykamp et al. [4].

of storage in electricity girds do not consider the potential shift of electricity demand during the day. This notion is further discussed in Chapter 9.

4.3 A Complete Approach to Energy Storage Algorithms

In this study we will extend the work of the aforementioned publications on revenue-oriented grid enhancement using storage. We will attempt to design and evaluate strategies for creating charge schedules using different scenarios. Our approach will extend the work of these publications in the following ways:

- Our work will focus on the Netherlands, by using market data from the Dutch electricity markets as load demand data from Dutch residential and commercial areas.
- In contrast to the separate evaluation of market prices in Nykamp et al. [4], our calculations will integrate two electricity markets, including the specific rules and time-dependent policies regarding each market.
- We will evaluate the costs and revenues using an online simulation that uses realistic forecasts, based on the quality of published forecasting studies. Furthermore, our approach will include stochastic events such as cable breakages to more realistic model an electricity grid.
- Our model for cable damage (Section 3.1) is capable of reflecting cable damage that agrees with recent publications.
- Our evaluation will focus on several situations in which storage can be used, using residential and commercial load demand data as well as distributed solar- and wind-energy energy generation.
- Our data collection and preparation steps include the ability to create repetitions through which we can asses the statistical significance of performances between different methods of creating charge schedules.
- In addition to using a single battery, we will consider a distributed approach in which multiple batteries protect a grid asset.

In this chapter we have introduced Grid Enhancement, Peak-Shaving and recent efforts in combining revenue-oriented and cable protection strategies using energy storage. In the next chapter, we will asses the theoretical model combining these two approaches. Later, we will introduce a simulation model to evaluate the performance of various algorithms and heuristics.

Chapter 5

Algorithms for Optimizing Grid Enhancement and Energy Trading through Energy Storage

Contents

In the previous chapter, we demonstrated how energy storage can be an effective way of enhancing electricity grids that experience peak demands. Thus, storage units such as batteries and may present an alternative to traditional grid reinforcements. Whether or not introducing storage units represent an economically favourable solution depends on the cost, lifetime and efficiency of the storage units, as compared to the cost of traditional reinforcements.[4, 2]

In this chapter, we introduce another way of lowering storage costs, by trading electricity on wholesale markets. By buying (storing) power at low prices, and selling (discharging) at higher electricity prices, storage units can generate money, while at the same time executing their task of grid enforcements. The storage unit can even buy and sell electricity at different markets.

By adding energy trading to the grid enhancement objective in our model, we created a multi-objective optimisation problem. At first, we deal with this by converting the grid enhancement objective to a constraint, while optimising the energy costs. Later, we will relax the grid enhancement constraints by using penalties in the optimisation objective. The inclusion of energy trading in our model requires the specification new problem definitions and objective functions. This chapter introduces algorithm to deal with these new problem definitions.

5.1 Energy Trading using Electricity Markets

When storage units are used for grid enhancements (e.g. peak-shaving), they have to be charged and discharged with electricity according to demand patterns and properties of the different entities (cables, transformers etc.) in the network. In the Netherlands (among many other countries), electricity can be bought at wholesale electricity markets. These markets exist for large-scale producers and consumers of electricity to bid against each other. This way, market prices are established while matching the in- and out-flow of electricity in the network. As a result, electricity prices fluctuate over time depending on the demand and supply of electricity (Section 2.2).

Since a storage unit charges and discharges electricity at different times, the storage unit is *de facto* trading electricity. There is a loss of energy due to suboptimal electricity, but as long as the difference in prices between bought and sold electricity is larger, the storage unit can make a profit.

In the Netherlands there exist multiple electricity markets. Therefore, electricity can be bought and sold at different markets. In economics, this way of trading is defined as *arbitrage* [44]. A controller of a storage unit does not even need to use the storage unit to exercise arbitrage. In the case where an amount of electricity was bought at the APX day-ahead market, the electricity can be sold on another market before the storage has to charge or discharge the electricity. Having the storage unit can therefore be used as a back-up to arbitrage trading, increasing the value of a storage unit. Such advanced tactics are not explicitly discussed later on, but the ability to buy and sell electricity at different electricity markets for a single interval does arise, and is part of optimal solutions to the problems defined below.

5.2 Simple Grid Enhancement & Energy Trading through Storage Problem (SGEETSP)

Here, we introduce the Simple Grid Enhancement & Energy Trading through Storage Problem (SGEETSP). The objective of this new model is to minimise the total cost of electricity, while supplying the necessary grid enhancements to prevent grid damage. For a planning horizon of T-intervals ($1 \leq t \leq T$) we minimise the function:

$$K(\{x_1, ..x_T\}) = \frac{1}{4}\sum_{t=0}^{T} f_t x_t \tag{5.1}$$

Here, x_t specifies the charge/discharge rate (kW) of battery B at interval t (negative and positive values indicate discharging and charging the battery, respectively). In this model we use 15-minute intervals. f_t indicates the electricity price at time t (€/kWh). As a result, the sum of charge/discharge rates is divided by 4 in (5.1). The storage decisions in this model are subject to three sets of constraints:

$$-P^c_{\max} \leq x_t + P^W_t \leq \; P^c_{\max}, \qquad 1 \leq t \leq T \tag{5.2}$$

$$P^B_{\min} \leq x_t \leq \; P^B_{\max}, \qquad 1 \leq t \leq T \tag{5.3}$$

$$0 \leq C^B_0 + \frac{1}{4}\sum_{k=0}^{t} x_k \leq \; C^B_{\max}, \qquad 1 \leq t \leq T \tag{5.4}$$

The maximum cable capacity $P^c_{\max}$ (kW) is satisfied using equation (5.2). As introduced in Chapter 4, cable c is utilized for connecting both connection W and battery B to the external grid. Hence, the sum of x_t and the power demand from W, P^W_t (kW) should lie within $[-P^c_{\max}, P^c_{\max}]$. (5.3) assures that the charge/discharge rate of battery B is within its allowed interval $[P^B_{\min}, P^B_{\max}]$. Finally, equation (5.4) guarantees the charge state of the battery (kWh) is always between zero and the maximum battery capacity $C^B_{\max}$. Here, $C^B_0 (0 \leq C^B_0 \leq C^B_{\max})$ refers to the initial charge of B.

This model is a simplified view on the entities in the model and also contains some strong assumptions. The battery is considered to have a 100% round-trip efficiency. This is not possible in real life, but this assumption allows us to consider related problems in the next section more easily. Later, this assumption will be removed.

Secondly, the battery is assumed to be able to resolve all potential grid infrastructure problems. For instance, given an input that contains the values $P_t^W = 500\text{kW}, P_{\max}^c = 100\text{kW}, [P_{\min}^B, P_{\max}^B] = [-50\text{kW}, 50\text{kW}]$, battery B will never be able to satisfy constrain (5.4), regardless of the battery charge at t. In order to handle such an input, our model should handle exceeding cable-induced constraints as well. These additions, among others, are added to the problem definition in Section 5.4.

5.2.1 A Pseudo-Polynomial Algorithm for Integer-valued SGEETSP

Consider the SGEETSP problem defined above with only integer values for $P_{\min}^B, P_{\max}^B, P_t^W, C_0^B, C_{\max}^B$, where only integer solutions for x_i are allowed. An optimal solution can be found by a straightforward dynamic programming algorithm. In order to find an optimal solution, we explore the state-space indicated by

$$F_i^k = \text{minimal cost of energy to reach time } t = i \text{ at a battery charge of } C_i^B = k \qquad 0 \leq i \leq T, 0 \leq k \leq C_{\max}^B \tag{5.5}$$

We define the following recurrence equation:

$$F_0^k = \begin{cases} 0 & k = C_0^B \\ \infty & \text{otherwise} \end{cases} \tag{5.6}$$

$$F_{i+1}^k = \min_{j \in J_{i,k}} \left[F_i^{k-\frac{1}{4}j} + \frac{1}{4} j f_i \right] \tag{5.7}$$

where

$$J_{i,k} = \{ \ j \mid P_{\min}^B \leq j \leq P_{\max}^B \wedge -P_{\max}^c \leq P_i^W + j \leq P_{\max}^c \wedge -4k \leq j \leq 4(C_{\max}^B - k) \} \tag{5.8}$$

In other words, the only allowed state at the start of the planning horizon is the initial battery charge C_0^B. We recursively find the optimal solution by iterating over all permitted charging rates $J_{i,k}$ at interval i and battery charge C_k^B. $J_{i,k}$ is derived from (5.3), (5.2) and (5.4)).

Proof of Optimality

Equations (5.6, 5.7) form the Bellman property of our problem. In order to show that using these equations we can find the optimal solution to the integer-valued SGEETSP we formulate the proof of optimality here.

Theorem 5.2.1. *F_{i+1}^k is the minimal cost of energy to reach a battery charge k at $t = i$.*

Proof. Proof by cases:

- *Case i=0*: For $i = 0$ the only allowed charge state of the battery is C_0^B, the charge of the battery at the start. No electricity has been bought/sold at this cost, so $F_{C_k^B}^0 = 0$. For all other values $k \neq C_0^B$, the state is unreachable and the costs are ∞.
- *Case i+1*: The minimal cost of reaching battery charge k at $t = i + 1$ can be reached by charging the battery for j energy in the interval between i and $i+1$. Thus, the optimal value for F_{i+1}^k consists of the cost of charging j and the cost of reaching $F_i^{k-\frac{1}{4}j}$ for some value j. If F_i^k $0 \leq k \leq C_{\max}^B$ are the optimal values for reaching all charge states k at $t = i$, it follows that $F_{i+1}^k = \min_{j \in J_{i,k}} \left[F_i^{k-\frac{1}{4}j} + \frac{1}{4} j f_i \right]$.

By induction, it holds that F_i^k is the minimal cost of energy to reach a battery charge k at $t = i$.

□

Algorithm

Definitions (5.6, 5.7) can be used to create a dynamic programming algorithm by computing all values F_i^k for $0 \leq i \leq T, 0 \leq k \leq C_{\max}^B$:

Algorithm 1 Dynamic programming algorithm for SGEETSP

for $i = 0$ to T **do**
  for $k = 0$ to $C^B_{\max}$ **do**
    $F^k_i \leftarrow \infty$ {Initialize States}
  end for
end for
$F^{C^B_0}_0 \leftarrow 0$ {Start state, initial battery charge}
for $i = 1$ to T **do**
  for $k = 0$ to $C^B_{\max}$ **do**
    for $j = P^B_{\min}$ to $P^B_{\max}$ **do**
      if j is permitted **then** {See (5.8)}
        $F^k_i \leftarrow \min(F^k_i, F^{k-\frac{1}{4}j}_{i-1} + jf_i)$ {Recursively optimise values}
      end if
    end for
  end for
end for
$F \leftarrow F^0_T$
for $k = 1$ to $C^B_{\max}$ **do**
  $F \leftarrow \min(F, F^k_T)$
end for
return F

Complexity

For each state F^k_i we iterate over all permitted charge rates. Worst case, these are $(P^B_{\max} - P^B_{\min})$ charge rates, regardless of equations (5.2, 5.4). To check whether a charge rate is permitted requires constant time. The running time of this algorithm is therefore $\mathcal{O}(TC^B_{\max}[P^B_{\max} - P^B_{\min}])$. The algorithms therefore requires polynomial time in terms of the values of the input. As long as the battery capacity as well as the allowed charge rates stay relatively low, this algorithm can be used to find the optimal charge decisions.

Since the algorithm requires polynomial time in the value, and not the size of the input, the algorithm is a pseudo-polynomial time algorithm. In the next section, we show how we can find a solution in polynomial time. The purpose of first introducing this dynamic programming here will be made clear when discussing the advanced version of the SGEETSP later on.

5.3 SGEETSP as a Min Cost Flow Problem

We can alternatively model the SGEETSP as a *min cost flow problem*. This provides a model that is solvable in (strongly) polynomial time. Therefore, we first introduce the min cost flow problem.

5.3.1 min cost flow problem

Consider a directed graph $G = (V, E, c, k)$ where V is a set of vertices, E as set of directed edges, and each edge $(u, v) \in E$ has a capacity $c(u, v) \in \mathcal{R}$ and cost $k(u, v) \in \mathcal{R}$. Edge costs are symmetrical, i.e. $k(u, v) = -k(v, u)$.

A feasible flow in G is a function $f : V \times V \to \mathcal{R}$ satisfying:

- Capacity constraints:
$$\forall (u, v) \in E : f(u, v) \leq c(u, v) \tag{5.9}$$
- Skew symmetry:
$$\forall (u, v) \in E : f(u, v) = -f(u, v) \tag{5.10}$$

- Balance constraints:
$$\forall u \in V : \sum_{v \in V \setminus \{u\}} f(u,v) = \sum_{v \in V \setminus \{u\}} f(v,u) \tag{5.11}$$

Capacity constraints $c(u,v)$ are allowed to be positive as well as negative. This allows for defining lower bounds on a flow between vertices. For a detailed explanation refer to Miltersen [45].

The cost of a flow f is defined as

$$\text{cost}(f) = \frac{1}{2} \sum_{(u,v) \in E} k(u,v) f(u,v) \tag{5.12}$$

The min cost flow problem is given a graph $G = (V, E, c, k)$, find a feasible flow f in G that minimizes the cost(f) among all feasible flows f in G. [45]

5.3.2 SGEETSP as a min cost flow problem

We model the SGEETSP as a min cost flow problem, by using flow as energy (kWh). An instance of a flow network that models a SGEETSP instance is shown in Figure 5.1. The graph G consist of vertices (one for each interval) and edges that represent the transfer of energy between intervals, either through stored or newly charged energy. As flow models energy in the flow network, we rewrite equations (5.3), (5.2) such that x_t is the amount of energy charged or discharged at interval t :

$$\begin{aligned} l_t \leq x_t \leq & h_t \\ \text{where } l_t = \max(\frac{1}{4}P^B_{\min}, \frac{1}{4}(-P^c_{\max} - P^W_t) &= \frac{1}{4}\max(P^B_{\min}, -P^c_{\max} - P^W_t) \\ h_t = \min(\frac{1}{4}P^B_{\max}, \frac{1}{4}(P^c_{\max} - P^W_t) &= \frac{1}{4}\min(P^B_{\max}, P^c_{\max} - P^W_t) \end{aligned} \tag{5.13}$$

Now, x_t is the charge rate in energy (kWh) in each interval. This removes the need for multiplying and dividing by 4 in the problem definitions. l_t and h_t now incorporate both the technical constraints of $P^B_{\min}$, $P^B_{\max}$ and $P^c_{\max}$ as well as the flexible demand on the cable as caused by P^W_t.

We can now construct a flow network that model the SGEETSP. The network G consists of:

- Vertices I, O.
- For each interval $t, 0 \leq t \leq T$, a vertex v_t.
- Edge $(v_t, v_t+1), 0 \leq t \leq T-1$ between consecutive time intervals, which models the transfer of charged energy between intervals. The edge capacities $c(v_t, v_{t+1}) = C^B_{\max}$,$c(vt+1, v_t) = 0$ constrain this flow within feasible battery capacity. The cost of this transfer is $k(v_t, v_{t+1}) = 0$.
- Edge $(I, v_t), 1 \leq t \leq T$ that represent charging the battery through the network. The lower and upper bounds of these flows are l_t, h_t, respectively. Therefore, $c(I, v_t) = h_t$ and $c(v_t, I) = -l_t$. The cost of a flow unit represents the price, hence $k(I, v_t) = f_t$. Sending flow from I to v_t is equivalent to charging the battery. A negative flow from I to v_t (a positive flow from v_t to I) is equivalent to discharging the battery.
- An edge (I, v_0) that represents the starting charge of the battery. Hence, this flow is constrained to C^B_0: $c(I, v_0) = C^B_0$, $c(v_0, I) = -C^B_0$. There is no cost for this flow, as the initial battery charge does not affect the energy cost ($k(I, v_0) = 0$).
- Edges (v_T, O) and (O, I) that connect that remaining flow with the "source" of flow I. The remaining charge of the battery in SGEETSP after interval t_T is effectively sold at a zero price. Therefore, these edges have costs $k(v_T, O) = 0$ and $k(O, I) = 0$. Only a positive flow is allowed from v_T to I, hence we have the lower bounds $c(O, v_T) = 0$ and $c(I, O)$.

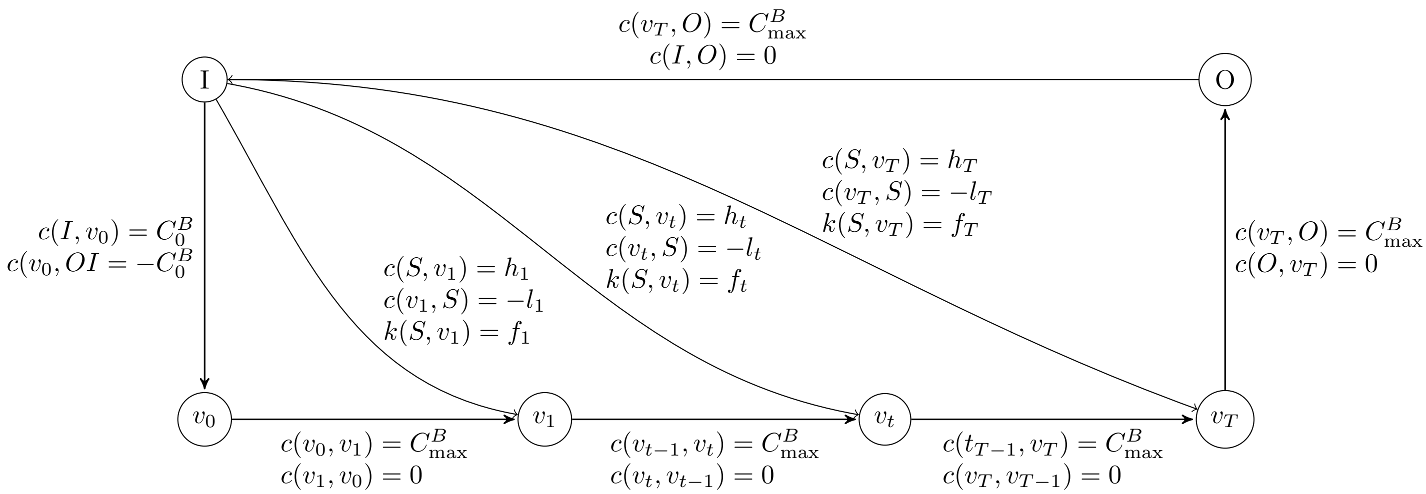


Figure 5.1: A min cost flow network for SGEETSP. For edges $(u, v) \in E$ where $k(u, v)$ is not shown, $k(u, v) = 0$.

The flow f of this network represents the flow of energy through the SGEETSP problem. Given a flow f, we can obtain a solution for SGEETSP.

Proof

Here, we show why the flow network G defined above can be used to solve SGEETSP. First, we show that the SGEETSP and min cost flow problem solutions can be used to find the related problems.

Theorem 5.3.1. *For an instance of SGEETSP and a derived graph G, if a feasible flow f in G exists, a feasible solution X in SGEETSP exists such that $L(f) = K(X)$.*

Proof. In order for a solution X to be feasible in SGEETSP, it needs to satisfy constraints (5.4) and (5.13). Let X be $\{x_t = f(I, v_t) | 1 \leq t \leq T\}$. Since $c(I, v_t) = h_t$, $c(v_t, I) = -l_t$ and $f(I, v_t) = -f(v_T, I)$, we have

$$-c(v_T, I) \leq f(I, v_t) \leq c(I, v_T) \tag{5.14}$$

$$l_t \leq f(I, v_t) \leq h_t) \tag{5.15}$$

$$l_t \leq x_t \leq h_t \tag{5.16}$$

Therefore, the solution X satisfies constraint (5.13). For every edge (v_{t-1}, v_t),$1 \leq t \leq T$ we have

$$-c(v_t, v_{t-1}) \leq f(v_{t-1}, v_t) \leq c(v_{t-1}, v_t) \tag{5.17}$$

$$0 \leq f(v_{t-1}, v_t) \leq C^B_{\max} \tag{5.18}$$

From the capacity and balance constraints it follows that

$$f(I, v_0) = C^B_0 \tag{5.19}$$

$$f(v_0, v_1) = C^B_0 \tag{5.20}$$

We derive $f(v_t, v_{t+1})$ using the balance constraint on vertex v_t:

$$\sum_{v \in V \setminus \{v_t\}} f(v_t, v) = \sum_{v \in V \setminus \{v_t\}} f(v, v_t) \tag{5.21}$$

$$f(v_t, v_{t-1}) + f(v_t, I) + f(v_t, vt+1) = f(v_{t-1}, v_t) + f(I, v_t) + f(v_{t+1}, v_t) \tag{5.22}$$

$$f(v_t, v_{t+1}) = f(v_{t-1}, v_t) + f(I, v_t) \tag{5.23}$$

$$= f(v_{t-1}, v_t) + x_t \tag{5.24}$$

Equation (5.24) is a recursive definition that can be solved using (5.20):

$$f(v_t, v_{t+1}) = f(v_0, v_1) + \sum_{k=1}^{t-1} [x_k] + x_t \tag{5.25}$$

$$= C_0^B + \sum_{k=1}^{t} x_t \tag{5.26}$$

If we combine (5.18) and (5.26), we have

$$0 \leq C_0^B + \sum_{k=1}^{t} x_t \leq C_{\max}^B \tag{5.27}$$

Therefore, solution X satisfies constraint (5.13). Since X satisfies both (5.4) and (5.13), X is a feasible solution in X. The cost $L(f)$ of flow f in G is given by

$$L(f) = \frac{1}{2} \sum_{(u,v)\in E} k(u,v)f(u,v) \tag{5.28}$$

$$= \frac{1}{2}\left(\sum_{t=1}^{T} k(I,v_t)f(I,v_t) - (k(I,v_t)(-f(I,v_t))\right) \tag{5.29}$$

$$= \sum_{t=1}^{T} k(I,v_t)f(I,v_t) \tag{5.30}$$

$$= \sum_{t=1}^{T} f_t f(I,v_t) \tag{5.31}$$

The cost $K(X)$ of X in SGEETSP is given by

$$K(X) = \sum_{t=1}^{T} f_t x_t \tag{5.32}$$

$$= \sum_{t=1}^{T} f_t f(I,v_t) \tag{5.33}$$

$$= L(f) \tag{5.34}$$

□

Theorem 5.3.2. *For an instance of SGEETSP and a derived graph G, if a feasible solution X in SGEETSP exists, a feasible flow f in G exists such that $K(X) = L(f)$.*

Proof. Let f be a flow in the derived network G:

$$f(I, v_0) = f(v_0, v_1) = C_0^B \tag{5.35}$$

$$f(I, v_t) = x_t \tag{5.36}$$

$$f(v_{t-1}, v_t) = C_0^B + \sum_{k=1}^{t-1} x_k \tag{5.37}$$

$$f(v_T, O) = f(O, I) = C_0^B + \sum_{t=1}^{T} x_t \tag{5.38}$$

Since none of the flows in f are directly conflicting, the skew symmetry constraints (5.10) are satisfied. In order for f to be feasible, it needs to satisfy the capacity and balance constraints (5.9) and (5.11). Therefore, we first show that for every edge in G, f satisfies the capacity constraints. Edges with identical capacity constraints and flows have been grouped together:

- (I, v_0), (v_0, v_1):

$$-c(v_0, I) = -c(v_1, v_0) \leq f(I, v_0) = f(v_0, v_1) \leq c(I, v_0) = c(v_0, v_1) \tag{5.39}$$
$$C_0^B \leq C_0^B \leq C_0^B \tag{5.40}$$

- (I, v_t):

$$-c(v_t, I) \leq f(I, v_t) \leq c(I, v_t) \tag{5.41}$$
$$l_t \leq x_t \leq h_t \tag{5.42}$$

- (v_{t-1}, v_t):

$$-c(v_t, v_{t-1}) \leq f(v_{t-1}, v_t) \leq c(v_{t-1}, v_t) \tag{5.43}$$
$$0 \leq C_0^B + \sum_{k=1}^{t-1} x_k \leq C_{\max}^B \tag{5.44}$$

- $(v_T, O), (O, I)$:

$$-c(O, v_T) = -c(I, O) \leq f(v_T, O) = f(I, O) \leq c(v_T, O) = c(O, I) \tag{5.45}$$
$$0 \leq C_0^B + \sum_{t=1}^{T} x_t \leq C_{\max}^B \tag{5.46}$$

Here, (5.42), (5.44), (5.46) hold since X is a feasible solution in SGEETSP. Next, we show that for every vertex $v \in V$ the balance constraint is satisfied:

- v_0:

$$f(I, v_0) = f(v_0, v_1) \tag{5.47}$$
$$C_0^B = C_0^B \tag{5.48}$$

- v_t, $1 \leq t \leq T - 1$:

$$f(v_{t-1}, v_t) + f(I, v_t) = f(v_t, v_{t+1}) \tag{5.49}$$
$$C_0^B + \sum_{k=1}^{t-1} [x_k] + x_t = C_0^B + \sum_{k=1}^{t} x_k \tag{5.50}$$
$$C_0^B + \sum_{k=1}^{t} x_k = \tag{5.51}$$

- v_T:

$$f(v_{T-1}, v_T) + f(I, v_T) = f(v_T, O) \tag{5.52}$$
$$C_0^B + \sum_{k=1}^{T-1} [x_k] + x_T = C_0^B + \sum_{t=1}^{T} x_t \tag{5.53}$$
$$C_0^B + \sum_{t=1}^{T} x_t = \tag{5.54}$$

- O:

$$f(v_T, O) = f(O, I) \tag{5.55}$$
$$C_0^B + \sum_{t=1}^{T} x_t = C_0^B + \sum_{t=1}^{T} x_t \tag{5.56}$$

- I:

$$f(I, v_0) + \sum_{t=1}^{T} f(I, v_t) = f(O, I) \tag{5.57}$$
$$C_0^B + \sum_{t=1}^{T} x_t = C_0^B + \sum_{t=1}^{T} x_t \tag{5.58}$$

Flow f satisfies all capacity, skew and balance constraints and is therefore a feasible flow in G. The cost $K(X)$ of X is given by:

$$\begin{aligned} K(X) &= \sum_{t=1}^{T} f_t x_t && (5.59)\\ &= \sum_{t=1}^{T} f_t f(I, v_t) && (5.60)\\ & && (5.61) \end{aligned}$$

The cost $L(f)$ of f is given by:

$$\begin{aligned} L(f) &= \frac{1}{2} \sum_{(u,v)\in E} k(u,v) f(u,v) && (5.62)\\ &= \frac{1}{2}\left(\sum_{t=1}^{T} k(I, v_t) f(I, v_t) - (k(I, v_t)(-f(I, v_t)) \right) && (5.63)\\ &= \sum_{t=1}^{T} k(I, v_t) f(I, v_t) && (5.64)\\ &= \sum_{t=1}^{T} f_t f(I, v_t) && (5.65)\\ &= K(X) && (5.66) \end{aligned}$$

□

Using Theorem 5.3.1 and Theorem 5.3.2 we can now prove that the min cost flow problem can be used to solve the SGEETSP:

Theorem 5.3.3. *For an instance of SGEETSP, if we have a derived graph G and a feasible flow f in G of cost $L(f)$, such that $L(f)$ is the minimal $L(f)$ among all feasible flows f in G, the optimal solution X of SGEETSP has cost $K(X) = L(f)$.*

Proof. From Theorem 5.3.1 it follows that there exist a solution X of SGEETSP such that $L(f) = K(X)$. Assume X is not the optimal solution of SGEETSP, therefore a feasible solution X' exists such $K(X') < K(X)$. From Theorem 5.3.2 it follows that since X' is a feasible solution in SGEETSP, there must also be a flow f' in G for which $K(X') = L(f')$. Since $K(X') < K(X)$, $L(f') < L(f)$ which would make f non-optimal. This is contradictory, hence X is the optimal solution of SGEETSP of cost $K(X) = L(f)$.

□

5.3.3 Algorithm, Complexity and real/integer-valued SGEETSP

Graph G consists of $T+3$ vertices and $2T+3$ edges. The complexity of finding the min cost flow in G depends on the algorithm used to solve the problem. Various algorithms for finding min cost flows exist. For instance, the minimum mean cycle cancelling algorithm by Goldberg and Tarjan [46] finds the optimal solution of any min cost flow problem in in $\mathcal{O}(nm^2 \log n)$, where n is the number of vertices V and m the number of edges. This algorithm could thus solve an instance of SGEETSP in $\mathcal{O}(T^3 \log T)$.

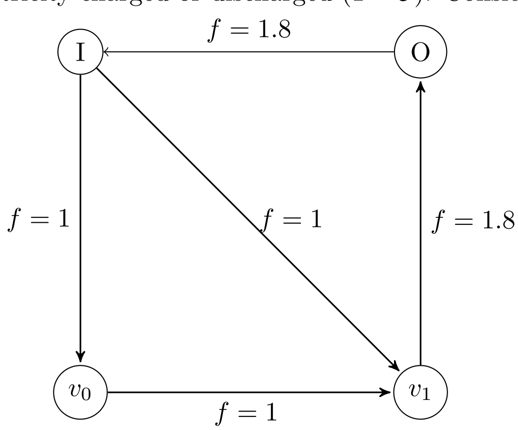
The min cost flow problem as well as the minimum mean cycle cancelling algorithm can be used for solving real-valued instances of SGEETSP. In Section 5.2.1 we considered the integer-valued version of SGEETSP. Doing so allowed us to show that an optimal solution could be found in pseudo-polynomial time using dynamic programming. The min cost flow problem can also be used for finding integer solutions, by using the Integrality Theorem:

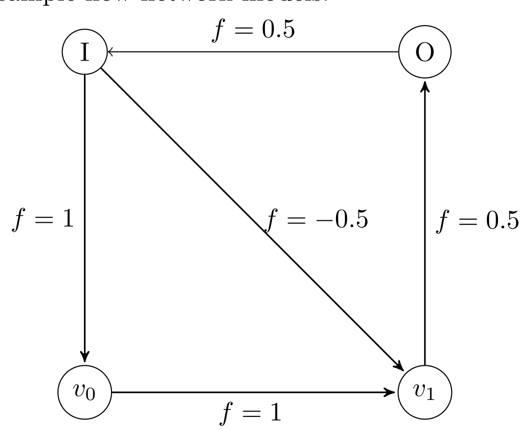
Theorem 5.3.4. *If* $\forall\ (u,v) \in E$*:* $c(u,v) \in \mathbb{N}$*,* $k(u,v) \in \mathbb{N}$*, there there exists a minimal cost flow* f *such that* $\forall\ (u,v) \in E:\ f(u,v) \in \mathbb{N}$.

Thus, an algorithm that solves the min cost flow problem to optimality will find an integer solution if all edge capacities and edge costs are integer-valued as well [47]. Therefore, using a min cost flow problem to solve the SGEETSP instead of the previous dynamic programming algorithm results in running times that are independent of the *values* of the problem parameters. The min cost flow solution therefore is an improvement on both the integer- and real-valued SGEETSP.

In the following section, we will extend the SGEETSP to include some crucial properties for using the algorithm on real-world applications. Some of these properties make the common min cost flow problem unusable for solving these new problem instances. For instance, we will include the efficiency of the battery (as discussed in Section 3.2) in the problem definition.

As a result, the flow network approach becomes unusable for this kind of problem. One of the fundamental properties of flow networks is the balance constraint (5.11). If we consider the efficiency of the battery, some of the electricity or flow is lost. Therefore we have to remove the balance constraint, or explicitly model the loss of electricity. Both approaches are not feasible. The balance constraint in integral to algorithms that work on flow problems. Removing this constraint on flow networks therefore disables us from utilising theorems and algorithms. Modelling SGEETSP as a flow problem then becomes useless. The second possibility where we would model the loss of electricity is also problematic. The amount of electricity lost is a fraction of the electricity charged or discharged $(1-\mathcal{E})$. Consider these two sample flow network models:





On the left, a $\mathcal{E} = 0.8$ battery is charged with one unit of electricity. The battery effectively is charged with 0.8 unit. On the left, the battery discharges 0.5 unit electricity. The efficiency doesn't count towards this efficiency and 0.5 is removed from the effective battery charge. Note that the graph on the right satisfies the flow graph balance constraint, while the left graph doesn't at node v_1.

We cannot force some amount of flow to leave the original graph at v_1 as the efficiency loss is a fraction of the electricity charged, and also dependent on whether we charge or discharge electricity. Other approaches that for instance adjust electricity prices and flow capacities also don't satisfy the efficiency property as changes reflect both charging and discharging electricity.

Among other properties, the imperfect efficiency of batteries causes the flow network approach to be unusable for finding charge schedules of real-world applications. However, the method provides an interesting demonstration how a problem can be modelled using flow networks. Furthermore, we hope this demonstration perhaps forms the basis of new graph algorithms that can be used for optimising electricity storage.

5.4 Grid Enhancement & Energy Trading through Storage Problem (GEETSP)

We now extend the SGEETSP in the following 5 ways:

1. As discussed in Section 2.1, the Dutch electricity system consists of multiple wholesale markets. We explicitly incorporate the APX and TenneT Imbalance markets in GEETSP. The APX market has a price f_t for every interval t for both buying and selling electricity. The TenneT Imbalance markets distinguishes two separate prices. These prices indicate whether a party adds or substracts electricity from the grid, the so-called up- or down-regulation. We denote these prices with $f_t^{\uparrow}$, $f_t^{\downarrow}$ as the up- and down-regulation prices (see Section 2.2.3). The amount of energy bought on each market is denoted by b_t, $b_t^{\uparrow}$, $b_t^{\downarrow}$ for the APX, up-regulation and down-regulation quantities, respectively. For an interval t, a solution can either up-regulate or down-regulate on the Imbalance Market, hence only one of $b_t^{\uparrow}$, $b_t^{\downarrow}$ can be non-zero. Furthermore, we limit the amount of electricity that can be bought at the APX market to the maximum discharge and charge rates, $P^B_{\min}$, $P^B_{\max}$.

2. In order to build useful charge planning with our algorithms, we define a target τ for the net amount of electricity charged during the planning window T. By setting τ, we can influence the charge level of the battery after T intervals. For instance, if we set $\tau = 0$, the charge level $C^B_T = C^B_0$.

3. In order to allow for a flexibility in the final level of the battery, we define a penalty p_S. for deviating from the target energy τ by 1 kWh. If for instance, the target τ is set to an infeasible value, the algorithm will not find a solution. By defining penalties we allow deviations from the target. Also, deviating from the set target τ might result in a financial or cable-damage preventing advantage. The value of the target-deviating penalty p_S reflects the economical preference.

4. Similar to the definition of target penalty, we define a cable damage penalty p_D. For every interval and every kW the cable capacity is exceeded, a penalty of p_D is counted. For every interval t, we denote the amount of power the cable is exceeded by as D_t. This allows solutions in which cable damage occurs to exists (in some cases, cable damage will be inevitable), and still to result in optimal battery schedules. Also, this allows us to specify the importance of cable damage financially. In some cases, cable damage might might be preferred over damage prevention for financial considerations.

5. In SGEETSP, the battery efficiency (the fraction of discharged energy per charged energy) was assumed to be 100%. Since batteries currently do not come close to this efficiency [43], the influence of the efficiency is of great influence on real-world charging decisions. We therefore introduce the efficiency $\mathcal{E}$ which reflects the round-trip efficiency.

Formally defining the new problem also provides a Mixed Integer Programming (MIP) formulation. This formulation is provided in Appendix F.1. This appendix also contains the introduction of an alternative dynamic programming algorithm that can be used to solve this program.

Chapter 6

An Online Evaluation of Algorithms & Strategies for Energy Storage in Electrical Networks

Contents

In this chapter we evaluate the performance of control strategies for batteries in electricity grids. This involves the algorithms discussed in the previous chapter, as well other types of control strategies. In order to evaluate these, we designed a simulation model (Section 6.2) that allows for dynamic calculations. Using three real-world-derived test scenarios (Section 6.3) we evaluate a collection of strategies (Section 6.4). The performance of these strategies is analysed in Section 6.5.

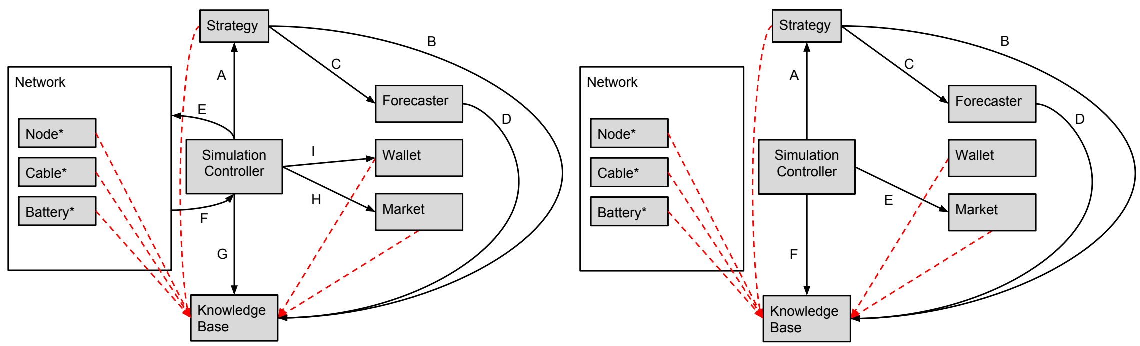


Figure 6.1: Simulation Entity Overview. **(A)** (*left*): Entities and messages in the Regular Interval event. **(B)** (*right*): Entities and messages in the Day-Ahead Planning event. Black arrows show method calls (directional), red arrows show information streams between entities and the simulation knowledge base. Description of the labeled method calls are provided in Section 6.2.2

6.1 Introduction

In the previous chapter we introduced models and algorithms that minimised electricity costs of storage (or maximised profits) while protecting electricity grids. These algorithms provide optimisations in deterministic cases; i.e. the node loads and electricity prices are known for the entire planning duration. In practice, this is evidently not the case as loads and prices depend on many external factors such as human behaviour, weather, market behaviour and so on.

To evaluate the potential benefits of implementing storage in electricity grids, our experiments should include this uncertainty about the future. In order to accurately evaluate how well storage control strategies work in practice, we designed a simulation model that incorporates uncertainty. This requires a control strategy to do two things: A strategy should (a) optimise profits and minimise grid damages using a planning horizon that approaches an long-term optimal solution while (b) adjusting previously made decisions to deal with changing loads and price patterns. In fact, it may not even be necessary to do (a) at all. In theory, a battery could be controlled using a control mechanism that only factors in current prices, battery properties and network loads, without maintaining a planning horizon as discussed in Chapter 5. In this chapter, we therefore distinguish *strategies* from *algorithms*. A strategy is a set of rules which makes decisions on the battery usage. As a part of this, the strategy may use an algorithm such as the DPGEET or MIPGEET algorithms from the previous chapter.

The dynamic nature of our evaluation leads us to create a simulation model which deals with changing forecasts, cable damage due to exceed the cable limitations and two separate electricity markets. In the following section we introduce this simulation model.

6.2 Simulation Model

The simulation model used in this chapter is a *discrete event simulation model*. Such a model uses a set of events to alter the state of a system, by having a set procedure to handle each event type. Each event in this model can change the state of batteries, cables, nodes, markets (among others). Furthermore, each event may also schedule one or more events in the future. An ordered queue is used to keep track of all remaining events, and to process the events chronologically in order. The state objects, or entities, as well as the events and associated protocols make up the simulation model.

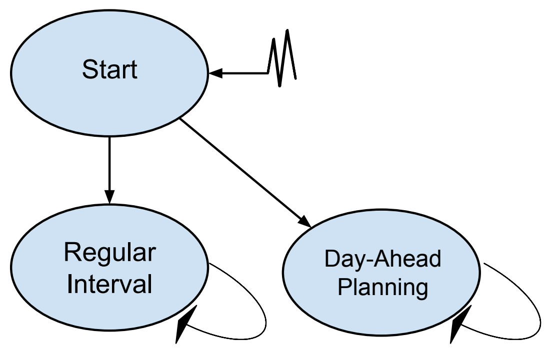


Figure 6.2: Simulation Event Graph.

6.2.1 Entities

The simulation model consists of the entities shown in Figure 6.1. At the center of the model, the Simulation Controller connects the information and decisions between different entities. The controller also keeps track of the event queue. When a decision is required, this controller queries the Strategy entity. The Strategy entity contains all intelligence relating to the battery control. This Strategy object can consult the Forecaster for predictions on load demands and electricity prices. When made, the decisions are transferred to the Network. The simulation model is flexible in this regard, such that the Network can be adjusted to the desired grid structure and cable, battery and node models. How a set of electrical loads affects the different entities is captured within the Network entity. The Network entity can also inform whether or not the decisions by the Strategy entity were possible, and how much electricity effectively was consumed/produced by the battery/batteries. The Market entity is informed of the effective electricity transfers and calculates the electricity costs. This entity also keeps track of the electricity market prices and bidding procedures. The electricity costs are reported to Wallet entity, which keeps track of the monetary balance. Finally, all objects have access to the Knowledge Base for storing and retrieving data regarding their state. Naturally, the Strategy object is prevented from accessing future values of load demands and electricity prices through the Knowledge Base.

By structuring the simulation model like this, parts of the simulation can easily be replaced, for instance if a different electricity market (with specific market rules) or battery model is preferred. Testing different battery control strategies only requires replacing the Strategy entity.

6.2.2 Events

There are three event types used in our simulation, one of which is the Start event which is used only once at the beginning of each simulation. This event is run to initialise all models and data sources. The Start event also schedules the first Regular Interval and Day-Ahead Planning events. The Regular Interval event takes place once per interval, which is this study is every 15 minutes. The Day-Ahead Planning event occurs daily, and queries the strategy for bidding on the APX market. An Event Graph of the simulation is shown in Figure 6.2. The procedure for the Regular Interval and Day-Ahead Planning events are explained below.

Regular Interval In any regular interval, there is a node load, as well as a charge rate of the battery. The Strategy entity is required to make a decision on the battery charge at the start of the interval. The actual node load, as well as the actual TenneT Imbalance prices are known to the Strategy entity. In Figure 6.1A the order of messages within the simulation are shown (A-F). These messages are:

A) the Simulation Controller queries the Strategy for battery charge rates.

B) the Strategy consults the Knowledge Base for historical and current properties and variables.

C) the Strategy consults the Forecaster for forecasts on unknown variables.

D) the Forecasters consults the Knowledge Base in order to produce forecasts.

E) the Simulation Controller informs the Network of the charge rate decisions.

F) the Network informs the Simulation Controller which charge rates actually occurred (a Strategy might make an unfeasible decision, this way the requested and actual charge rates will be different)

G) the Simulation Controller notifies the Knowledge Base to collect information regarding the past interval

H) the Simulation Controller reports the charge rates that were used and is presented the electricity cost by the Market.

I) the Simulation Controller informs the Wallet entity of the electricity costs.

This order of steps allows a single event to take care of the protocol, while building checks to see wether the Strategy decisions are feasible.

Day-Ahead Planning In the Day-Ahead Planning event, the Strategy is queried to make hourly APX bids for the next day. The event occurs at 10:00 in the morning, and requests 24 APX bids for the following day from 0:00 to 0:00. The messages (calls) in this event are:

A) the Simulation Controller queries the Strategy for APX bids

B) the Strategy consults the Knowledge Base for historical and current properties and variables.

C) the Strategy consults the Forecaster for forecasts on unknown variables.

D) the Forecasters consults the Knowledge Base in order to produce forecasts.

E) the Simulation Controller informs the Network of the APX bids

F) the Simulation Controller notifies the Knowledge Base to collect information regarding the APX bids

6.2.3 Grid Structures

In our simulation model we incorporated an abstract layer (Network entity) such that any electricity grid structure can be evaluated, including any number of batteries, cables and nodes. In this chapter, we only evaluate the simple Storage-Cable-Node model grid structure (Section 3.1). A discussion of more complex grid structures in provided in Section 8.2.

6.2.4 Extended Simulation Model

In the development of the simulation model, we first attempted to create a simulation model that included the breakage of cables. If a cable becomes excessively damaged due to heat, or the cable is damaged by external forces (e.g. construction work), the cable is incapable of transporting electricity. In some cases, the cable breakage could be caused by faulty control strategy. If this happens, a storage unit could allow a household, company or other installation to still consume/produce electricity for a finite amount of time. Therefore, we would like to test how an algorithm deals with such occurrences. In order to do so, we developed a more complex simulation model than the model presented before. This model included separate event types for expected (due to overheating) and unexpected (due to external factors) cable failures and subsequent repair times (Figure C.3). These event types and their handling are included in the source code of this project.

Ultimately, we decided not to include these events in the simulations. The data that was available for the experiments did not allow for sufficient simulation durations to realistically reflect the frequency in which these breakages occur. Instead, we use a different method to calculate the benefits of operating a damaged electricity grid with distributed storage. These calculations are presented in Section 8.3.

6.2.5 Forecasters

As discussed earlier, we include forecasting (and subsequent forecasting errors) in our simulation. We do so by forcing our algorithms to make decisions on forecasted values. At a later time the forecasted values will change, and the strategies are required to act again. For instance, a charging schedule for a battery that uses the grid at full capacity might be dangerous as forecasting errors are made, resulting in inevitable grid damage later on. Also, energy prices might be higher or lower compared to price forecasts, resulting in a different electricity cost.

Including the actual forecasting of load demands and energy prices is beyond the scope of this study. For a discussion on publications on load and price forecasting see Section 2.4. Here, we focus on the *performance* by forecasting methods, without specifying these models themselves. In the simulations all *real* values are known. For instance, we use data for the load demand of households, companies and other nodes. Additionally, we have APX and TenneT Imbalance prices for a wide range of years. Hence, we only need to produce forecasts base on real values. The idea, therefore, is to take the original data, and add noise that resembles typical forecasting errors. Different forecasters thus introduce different kinds of forecasting errors.

Forecasting Error Measurement

First, we need to choose a suitable error measurement. Nijhuis [12] features a discussion on different forecasting error measures for load demand (see Section 2.4). According to Nijhuis [12], the benefit of reporting Mean Absolute Scaled Error (MASE) is that this value behaves well close to zero, and can also be used for comparisons between data sets. Alternatives, such as Root Mean Square Error (RMSE) (penalises large values) and Mean Absolute Percentage Error (MAPE) (approaches infinity around close-to-zero values) do not fulfil these requirements. MASE scales forecasting errors to the average error made by a naive forecasting method:

$$\text{MASE} = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}\right) \tag{6.1}$$

Here, MASE is the forecasting error made on a series of n values Y_i. e_t is the forecasting error made on value t: $e_t = Y_t - F_t$, where F_t is the forecasted value for Y_t. Thus, MASE compares forecasting errors to forecasting errors made by the forecasting method $F_t = Y_{t-1}$. This forecast error measure allows us now to define the *goodness* of a forecaster from the literature. For instance, Nijhuis [12] compares forecasting methods on load demand by groups of households (*phases* in the low voltage network). For 18 different phases, their advanced methods result in MASE values ranging from 0.62 to 0.85. Note that these predictions are done on a 24 hour forecasting horizon. On average, a simple regression methods yields a MASE of 0.792 on a single phase.

We use 2 forecasters in this model. For the theoretical case where all values can be forecasted without any error, we define a *perfect* forecaster. If the forecaster is queried for a value, the real value is returned. The other forecaster, which will be referred to as the *realistic* forecaster, has a MASE of 0.7 for forecasting electricity demand. Based on the results from Nijhuis [12], we conclude that load demand can currently be realistically be forecasted at a MASE of 0.7. More elaborate forecasting models are discussed and analysed in Section 2.4.

The perfect forecaster predicts also the APX and TenneT Imbalance prices without any errors. For the realistic forecaster, we unfortunately couldn't find applicable literature on how precise the market prices (APX, TenneT Imbalance) in the Netherlands can be forecasted. As no knowledge on this precision is publicly available, we chose to use a naive forecasting method for the APX data: the price 24-hours prior to the requested value is taken as a forecast. The TenneT Imbalance market is known to behave chaotic and unpredictable. We therefore returned pessimistic prices when querying this forecaster for Imbalance price forecasts.

The purpose of using these forecasting models is to see the effects of the difference in the current and potential quality of forecasters. In Section 8.1 the quality of forecasters that's necessary for utilising planning algorithms is discussed. The MASE-forecasts were generated using a method presented in Section F.2.

6.3 Test Data & Scenarios

In order to evaluate battery control strategies, 3 scenarios were defined for the simulations in this chapter. These scenarios reflect various real-world situations in both the present and future. Scenarios A & B reflect the situation where a battery is used to balance the load demand from a residential and commercial area, respectively. Scenarios C reflects future scenarios in which storage for PV systems are introduced. In each of these scenarios we have at least one element that allows for variation between repetitions based on a random seed. This allows for testing the significance in the simulation outcomes between different algorithms and strategies using a paired t-test. The choice of battery and cable properties in each scenario are justified in Section 6.3.4.

6.3.1 Scenario A: Ettenleur Residential Area

Section 2.5.1 discusses the Ettenleur Smart Storage Unit (SSU), a real-world case of interest regarding electricity storage. In this scenario, we model this case by using a battery to balance the load of a small neighbourhood of 240 houses with a maximum total load demand of 385 kW. The battery has a maximum storage capacity of 230 kWh, a maximum charge speed of 400 kW and a maximum discharge speed of 100 kW. The round-trip efficiency of the battery has not been reported. We therefore assume that the battery has a 80% round-trip efficiency, a number used in publications by Ramezani et al. [3] and Nykamp et al. [4]. The cable connecting the Ettenleur neighbourhood to the external grid has a maximum power capacity $P^c_{\max} = 400kW$.

The neighbourhood is modelled using the *Zonnedael* data set (Section 3.4.1). As this data set consists of the 15-minute interval data for only 75 houses, we have to process this data on order to model the Ettenleur situation. A trivial way of doing this would be by scaling the sum of these houses by $\frac{240}{75} = 3.2$. However, this does not guarantee that the Ettenleur situation is suitable modelled in terms of maximum load demand. Instead, we randomly select and add residences from the *Zonnedael* set until the maximum load demand of our generated neighbourhood reaches 385 kW. This also introduces variations between repetitions of this scenario needed to test the significance of the outcomes. As a result, the number of houses in a repetition may be different, but the most critical grid usage will be equal. As the *Zonnedael* data was collected in 2013, we use the APX and TenneT Imbalance prices from 2013 as well. The scenario duration is one year ($0 \leq t \leq 35040$).

6.3.2 Scenario B: Schiedam Commercial Area

Scenario A reflects the use of a battery in a residential area. In order to evaluate the use of electrical storage in a non-residential area Scenario B models a commercial area. Commercial here refers to businesses, office buildings, small industries and other public services such as libraries and sport clubs. Section 3.4.1 describes the *Schiedam* data set, consisting of 69 municipal entities such as libraries, office buildings and sports facilities. In Scenario B we randomly select 20 out of the 69 entities and add their load demand together. This creates the variations between repetitions that allow for testing the significance of the outcomes. The 20 municipal entities represent a campus or business area. Similar to the Ettenleur Smart Storage Unit (SSU), we use a battery to balance the total load demand in the LV grid of a number of grid connections taken together.

The battery in each repetition has the same properties as the battery used in Scenario A. The only difference is that we scale the maximum battery capacity ($C^B_{\max}$) to the maximum load demand of the summed 20 entities. For example, if the peak demand over the course of one year is 1000 kW, we set the battery capacity to 1000 kWh as well. In other words, the battery is able to charge the maximum load demand for 1 hour. This ratio is twice that of the Ettenleur SSU case.

In Scenario A the cable has a higher maximum capacity (400 kW) then the maximum load demand (385 kW). Thus, use of the battery is not necessary for balancing the grid. In Scenario B, we set the cable maximum capacity ($P^c_{\max}$) to 85% of the peak demand. As a result, the battery will have to *balance* the grid at some point in order to prevent or minimise cable damage. The effects of different ratios between $P^c_{\max}$ and peak demand is analysed in Section 8.3.

Scenario B also occurs in 2013, and hence the APX and TenneT Imbalance prices from 2013 are used. The scenario duration is one year ($0 \leq t \leq 35040$).

6.3.3 Scenario C: Commercial Area with PV

Here, we investigate future developments with regards to implementing PV systems in commercial and residential settings. In the scenarios above there is minimal or none production of electricity. With the continued introduction of PV in both commercial and residential settings, the distributed production of energy introduced challenges in the entire electricity grid. In Scenario C we investigate to which extend using electricity storage can be profitable for owners of PV installations.

In order to test this, we need load demand data of PV installations. In many cases, these installations are coupled with another usage pattern, for instance residential home owners that consume electricity. Since we couldn't acquire such a data set, we use to separate data sources to simulate this data. First, we use EDSN profiles (Section 3.4.1) that are used in the industry as typical electricity usage patterns. Second, we add data for a PV installation by taking a resource with daily PV production values (Solar-Log, Section 3.4.2) and values per 15-minute intervals (Moraitis, Section 3.4.2).

The load demand of nodes in Scenario C are formed by combining EDSN profiles and the Solar-Log and Mortaitis PV data. For each node, we randomly select on one the of EDSN profiles. The total yearly electricity consumption of the node is randomly (uniformly) taken in the domain [2000,4000] kWh per year. The profile is scaled to this total yearly energy consumption. We then randomly select one of the 250 Solar-Log installations. This gives the node a daily energy production. To determine what the PV production per 15-minute interval is, we sample the Moraitis data. Given day k, we pick a day within 30 days from k in the Moraitis set. This may lead to some days be used more than once, but the advantage is that this process creates variation between repetition We scale the daily production to match the chosen Solar-Log production for that day, and add this profile to the EDSN-sampled consumption.

As Scenario C evaluates only the profitability of storage, no cable-induced constraints are used. In other words, the electricity grid surrounding the nodes is assumed to handle the peak demand without damaging the system. In the simulation model this is easily modelled by using cables with infinite cable maximum ($P^c_{\max} = \infty$).

The battery used in Scenario C is chosen depending on the average daily energy production per node. Earlier we defined P^W_t as the load demand by node W at time t in kW. The average daily energy production of a node is:

$$\text{Average Daily Production (kWh)} = \frac{1}{4}\frac{T}{96}\sum_{t=0}^{T} y_t \text{ , where} \tag{6.2}$$

$$y_t = \begin{cases} P^W_t & P^W_t < 0 \\ 0 & otherwise \end{cases} \tag{6.3}$$

In other words, we take all 15-minute intervals in which electricity is produced ($P^W_t > 0$) and see how much is produced on average. For nodes with an average daily production below 20 kWh, we assign a battery with capacity 20 kWh. For nodes with a daily production above 60 kWh, we assign a battery with 60 kWh capacity. For nodes with an average daily production between 20 kWh and 60 kWh, we assign a battery with a 40 kWh capacity. Each battery has charge speeds capable of charging/discharging the entire capacity in one hour and a 80% round-trip efficiency $\mathcal{E}$. Scenario C takes place in 2014. The APX prices and TenneT Imbalance prices for 2014 are used, as well as the EDSN profiles and Solar-Log data for 2014. The duration of this scenario is one year.

6.3.4 On the definition of battery and cable properties

Scenarios B and C are artificial and have no specific real-world situation they reflect. As a result, we have to define properties of the various entities ourselves. In Scenario B the battery properties are related to the node demand, as such a relatively large battery will be likely custom build in reality. In Scenario C we use fixed size batteries ($C^B_{\max} \in \{20 \text{ kWh}, 40 \text{ kWh}, 60 \text{ kWh}\}$) to choose for in each repetition, as this scenarios reflects the situation where a battery is offered as a large-scale product to customers.

To select the charge capacity of the batteries, we chose to use SSU-Ettenleur example as a standard. In that situation, a neighbourhood with maximum load demand 385 kW is modelled with a 230 kWh capacity battery. In this chapter, we therefore define batteries with a charge capacity capable of sustaining the maximum load demand for ∼30 minutes. In Scenario B, where the battery properties are custom, the charge capacity is thus directly scaled according to the maximum load demand of the node. In Scenario C a fixed set of batteries is defined. The battery with the nearest charge capacity to $\frac{1}{2} *$ *maximum load demand* is selected.

Scenarios B also includes a cable connecting the nodes and batteries to the external grid. In order to make sure the battery is needed for grid enhancement, we defined the maximum power capacity $P^c_{\max}$ of the cable connecting the entities to the grid at 85% of the maximum load demand. To investigate the profitability and applicability of electrical storage in a specific electricity grid, a system operator should first obtain these ratios before making these calculations.

6.4 Strategies

In the simulation model, strategies are required to make the decisions:

1. **ChargeDecision**: During each regular interval (Regular Interval event) a charge rate has to be established for the battery. At the time of the decision, the exact values of the market prices, load demands and cable states during that interval are known. For each battery, the strategy is queried for a charge rate. In case this charge rate is zero, the battery is not used during the interval. In some cases, the requested charge rate is not possible (for instance if the battery charge level is insufficient). In that case, the closest feasible charge rate is used.

2. **APXBidding**: Each day the battery is queried for placing bids on the APX markets. As the deadline for these bids is 10:00 am, the 40th (15-minute) interval is preceded by the Day-Ahead Planning event. In this event, the strategy is queried for an amount energy (kWh) and price (€/kWh) for each of the intervals occurring the next day. After this event, the bids are immediately awarded by the Market object. For future decisions, a strategy can therefore query the Market whether the bids were awarded or not. For a discussion on the APX bidding system, see Section 2.2.1.

Each strategy in this chapter has constains a unique combination of **ChargeDecision** and **APXBidding** decision procedures, although some strategies share one of the same. Before we introduce the set of strategies, we therefore first introduce the different APXBidding and ChargeDecision procedures. Note that these strategies are all aimed to work on a single storage unit. If multiple batteries are simulated, these strategies would create charge rates and APX bids for each storage separately, and sum the market and electricity usage.

6.4.1 APXBidding

The APXBidding decision is executed once a day, and allows strategies to plan the battery usage by securing energy on the APX market. The APXBidding decision requires the strategy to decide on a bid volume (kWh) and a bid price (€/kWh). The strategies below therefore return a tuple (bid volume, bid price). Please refer to Section 2.2.1 for an explanation of the APX Day-Ahead market bidding system. Note that bidding a very high price for electricity purchases (charging) ensures getting a bid awarded, and that bidding at price zero ensures selling electricity on the market.

In the following procedures, a query is made for the APX bid volume and price at interval X. These are the APXBidding procedures used in the strategies:

MIPGEET The MIPGEET algorithm (Section 5.4) can find an optimal solution for a battery incorporating both the TenneT and APX market prices. Strategies that use this algorithm remember both the TenneT and APX volumes. The APX volumes are immediately communicated to the Market, while TenneT volumes are remembered and potentially used in future ChargeDecisions.

This procedure contains parameters for the planning length in the MIPGEET algorithm (Duration) and the required cable and target penalties for the MIPGEET algorithm. The procedure uses a parameter ChargeLevel for which it aims the battery fill to be at the end of the schedules. Without this parameter, the MIPGEET algorithm is likely to produce schedules that end with empty batteries.

APXBidding-Procedure 1 MIPGEET(X, ChargeLevel, Duration, CablePenalty, TargetPenalty)

```
τ ← C_t^B − ChargeLevel
T ← Duration
p_D ← CablePenalty
p_S ← TargetPenalty
b, b↑, b↓, I, O ← MIPGEET(T, C_0^B, C_max^B, P_min^B, P_max^B, ℰ, P_max^c, P^W, f, f↑, f↓, τ, p_D, p_S)
if b_t > 0 then
  return (b_X, 0)
else
  return (b_X, ∞)
end if
```

DPGEET The DPGEET algorithm introduced in the previous chapter can be used similar to MIPGEET algorithm as described above.

APXBidding-Procedure 2 DPGEET(X, ChargeLevel, Duration, CablePenalty, TargetPenalty)

```
τ ← C_t^B − ChargeLevel
T ← Duration
p_D ← CablePenalty
p_S ← TargetPenalty
b, b↑, b↓, I, O ← DPGEET(T, C_0^B, C_max^B, P_min^B, P_max^B, ℰ, P_max^c, P^W, f, f↑, f↓, τ, p_D, p_S)
if b_t > 0 then
  return (b_X, 0)
else
  return (b_X, ∞)
end if
```

RHGEET The Ramezani et al. [3] heuristic discussed in Section 4.2.3 calculates an optimal battery usage for a single battery on one electricity market. This heuristic can also be used in our simulation model, if we only apply this algorithm to the APX market.

APXBidding-Procedure 3 RHGEET(X, Duration)

$T \leftarrow$ Duration
$\mathbf{b} \leftarrow$ RHGEET$(T, C_0^B, C_{\max}^B, P_{\min}^B, P_{\max}^B, \mathcal{E}, P_{\max}^c, \mathbf{P^W}, \mathbf{f})$
if $b_t > 0$ **then**
  return $(b_X, 0)$
else
  return (b_X, ∞)
end if

StochasticAPX In this procedure, a strategy has a ratio of charging versus discharging electricity and an APX bid price for charging and discharging electricity. The strategy randomly places bids on the APX markets. As explained in Section 6.2.2, the strategy only acquires electricity below or above it's bid prices for charging and discharging electricity, respectively. Using stochasticity is a way for a strategy to acquire cheap electricity, which can be sold on the TenneT Imbalance market later on.

APXBidding-Procedure 4 StochasticAPX(ChargeDischargeRatio, APXChargeVolume, APXChargePrice, APXDischargeVolume, APXDischargePrice)

$r \leftarrow$ random number between 0 and 1
if $r <$ ChargeDischargeRatio **then**
  return (APXChargeVolume, APXChargePrice)
else
  return (APXDischargeVolume, APXDischargePrice)
end if

None A strategy is not required to place bids on the APX market. This procedure purchases or sells no electricity on the APX market.

APXBidding-Procedure 5 None(X)

return $(0, 0)$

6.4.2 ChargeDecision

ChargeDecision procedures make a decision on the battery charge rate during each Regular Interval event. In the procedures below, the procedure is queried for a charge rate at interval t, which is the "current" interval in the simulation.

The following ChargeDecision procedures are used:

Memory-APXBidding [H] Strategies that use a planning algorithm can use the planning that was computed at the earlier APXBidding decision. As a forecast was used at earlier intervals, such a planning might be based on different market prices or load demands.

Procedure 1 Memory-APXBidding

$b, b^{\uparrow}, b^{\downarrow}, I, O \leftarrow$ results from previous MIPGEET/DPGEET/RHGEET procedure.
return $I_t - O_t$

CableGreedy CableGreedy decisions try to prevent cable damage in the interval. CableGreedy requires a parameter τ which is the desired battery fill (as a fraction $[0, 1]$). First the desired charge rate to achieve this fill is created (TCR). Then, the procedure uses the allowed battery charge rates and the interval allowed by the cable maximum to adjust the TCR to a feasible and least-damaging charge rate. This procedure does not evaluate electricity prices.

ChargeDecision-Procedure 2 CableGreedy(τ)

TCR $\leftarrow \frac{1}{4}\frac{\tau}{C^B_{\max}}$ {TargetChargeRate}
cableInterval $\leftarrow [-P^c_{\max} - P^W_t, P^c_{\max} - P^W_t]$
batteryInterval $\leftarrow [P^c_{\min}, P^c_{\max}]$
interval $\leftarrow$ INTERSECT(cableInterval, batteryInterval)
if interval = NULL **then**
    if $P^B_{\min} > (P^c_{\max} - P^W_t)$ **then**
        return $P^B_{\min}$
    else
        return $P^B_{\max}$
    end if
else if TCR $\in$ interval **then**
    return TCR
else if TCR $>$ interval.max **then**
    return interval.max
else
    return interval.min
end if

The sub-procedure INTERSECT() is used to find the overlap (if present) between to intervals:

Sub-procedure 1 INTERSECT(intervalA, intervalB)

if intervalA.max $<$ intervalB.min **or** intervalB.max $<$ intervalA.min **then**
    return NULL
else
    return [max(intervalA.min, intervalB.min), min(intervalA.max, intervalB.max)]
end if

PriceHeuristic The PriceHeuristic procedure uses the TenneT Imbalance price, the current battery charge, load demand and cable capacity maximum to form a decision on the charge rate. In short, the heuristic tries to prevent cable damage as much as possible, and charges or discharges the battery based on charge level and electricity prices. A relatively empty battery is charged unless the prices are dramatically high, while a relatively full battery is discharged, unless the prices are dramatically low. This heuristic is schematically shown in Figure 6.3.

Given a current battery charge and awarded APX volume, the new virtual charge level is computed if the APX bid were to be exactly followed. For each of three different segments of the battery, two prices indicate whether the battery should additionally charge or discharge more by comparing these prices ρ to the TenneT imbalance prices $f^{\downarrow}_t, f^{\uparrow}_t$. This interval is restricted by the cable maximum capacity. This heuristic provides incentives to keep the battery charge well between zero and maximum capacity, while utilising price spikes in the TenneT Imbalance market. This procedure is greatly affected by the chosen price thresholds ρ^+_{top}, ρ^-_{top}, ρ^+_{mid}, ρ^-_{mid}, ρ^+_{bottom}, ρ^-_{bottom}. Finding optimal values for these thresholds is further discussed in Chapter 7.

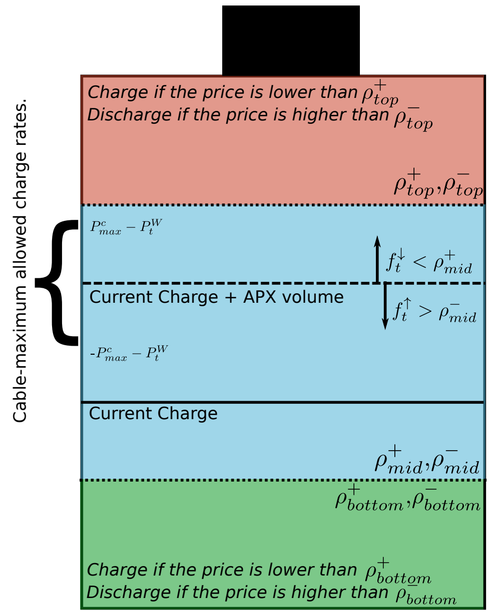


Figure 6.3: PriceHeuristic model for battery charge decisions.

Procedure 3 PriceHeuristic($\tau_{\text{lower}}, \tau_{\text{upper}}, \rho^{+}_{\text{lower}}, \rho^{-}_{\text{lower}}, \rho^{+}_{\text{mid}}, \rho^{-}_{\text{mid}}, \rho^{+}_{\text{upper}}, \rho^{-}_{\text{upper}}$)

cableInterval $\leftarrow [-P^{c}_{\max} - P^{W}_{t}, P^{c}_{\max} - P^{W}_{t}]$
batteryInterval $\leftarrow [P^{c}_{\min}, P^{c}_{\max}]$
interval $\leftarrow$ INTERSECT(cableInterval, batteryInterval)
if interval = NULL **then**
  if $P^{B}_{\min} > (P^{c}_{\max} - P^{W}_{t})$ **then**
    return $P^{B}_{\min}$
  else
    return $P^{B}_{\max}$
  end if
end if
VCR $\leftarrow 4\times$ APXAwardedVolume {virtual charge rate}
while true do
  VF $\leftarrow \frac{C^{B}_{t} + (4\text{VCR})}{C^{B}_{\max}}$ {virtual fill}
  if VF $< \tau_{\text{lower}}$ **then**
    VCR' $\leftarrow$ DOMAIN_VCR(0, τ_{lower}, VCR, VF, ρ^{+}_{lower}, ρ^{-}_{lower}, $f^{\uparrow}_{t}$, $f^{\downarrow}_{t}$, $C^{B}_{\max}$)
  else if VF $\geq \tau_{\text{lower}}$ **and** VF $< \tau_{\text{upper}}$ **then**
    VCR' $\leftarrow$ DOMAIN_VCR(τ_{lower}, τ_{upper}, VCR, VF, ρ^{+}_{mid}, ρ^{-}_{mid}, $f^{\uparrow}_{t}$, $f^{\downarrow}_{t}$, $C^{B}_{\max}$))
  else
    VCR' $\leftarrow$ DOMAIN_VCR(τ_{upper}, 1, VCR, VF, ρ^{+}_{upper}, ρ^{-}_{upper}, $f^{\uparrow}_{t}$, $f^{\downarrow}_{t}$, $C^{B}_{\max}$))
  end if
  if VCR = VCR' **then**
    if VCR $\in$ interval **then**
      return VCR
    else if VCR > interval.max **then**
      return interval.max
    else
      return interval.min
    end if
  end if
end while

The code of the PriceHeuristic perhaps appears complex. The sub-procedure DOMAIN_VCR (domain virtual-charge-rate) establishes the desired charge rate for one of the three segments of the battery. The PriceHeuristic procedure repeatedly calculates the desired charge rate (using DOMAIN_VCR) until no changes occur any more.

Sub-procedure 2 DOMAIN_VCR(L,H,VCR,VF,ρ^+, ρ^-, $f^\uparrow$, $f^\downarrow$, $C^B_{\max}$)

---

if $f^\uparrow \geq \rho^+$ **then**
  return $4(H - \mathrm{VF})C^B_{\max}$
else if $f^\downarrow \leq \rho^-$ **then**
  return $4(\mathrm{VF} - L)C^B_{\max}$
else
  return VCR
end if

MIPGEET-small Strategies can also utilise the MIPGEET algorithm during regular intervals. Strategies that do so use a shorter planning length compared to the APXBidding planning. The *target* charge (τ, Section 5.4) is derived from previous schedules. For example, if the MIPGEET is run for only the next 4 hours, the planning is required to finish the planning with the same battery charge as the battery charge would be 4 hours into the previous schedules. This guarantees the original, long-term schedule to be followed, while being to adjust to unexpected price or demand changes.

Procedure 4 MIPGEET-small(Duration, CablePenalty, TargetPenalty)

$T \leftarrow$ Duration
$\tau \leftarrow C^B_{t+T} - C^B_t$ according to previous MIPGEET schedule
$p_D \leftarrow$ CablePenalty
$p_S \leftarrow$ TargetPenalty
$\mathbf{b}', \mathbf{b}^{\uparrow\prime}, \mathbf{b}^{\downarrow\prime}, \mathbf{I}', \mathbf{O}' \leftarrow \mathrm{MIPGEET}(T, C^B_0, C^B_{\max}, P^B_{\min}, P^B_{\max}, \mathcal{E}, P^c_{\max}, \mathbf{P^W}, \mathbf{f}, \mathbf{f}^\uparrow, \mathbf{f}^\downarrow, \tau, p_D, p_S)$
Update $\mathbf{b}, \mathbf{b}^\uparrow, \mathbf{b}^\downarrow, \mathbf{I}, \mathbf{O}$ with the new values in $fb', \mathbf{b}^{\uparrow\prime}, \mathbf{b}^{\downarrow\prime}, \mathbf{I}', \mathbf{O}'$
return $I_t - O_t$

6.4.3 Strategies

Below, we introduce the strategies used in this chapter. We distinguish heuristic and optimisation strategies in this chapter. Optimisation strategies are strategies that make use of either the MIPGEET or DPGEET algorithms. Although the RHGEET heuristic also provides a planning, we consider it a heuristic strategy, as the produced planning is not guaranteed to be an optimal planning. Heuristic strategies don't compute the guaranteed optimal solution but use a heuristic to approximate the solution. Naturally, optimisation strategies rely much heavier on forecasts, as each demand and price of electricity should be known.

Each strategy makes has an APXBidding and a ChargeDecision procedure. Table 6.1 summarises the strategies given these two strategies. Below we will discuss the strategies. Note that the terms between parenthesis indicate the given APXBidding and ChargeDecision procedure.

Heuristic Strategies

H1 (None, CableGreedy) is the basic heuristic that attempts to to relieve grid issues whenever possible, regardless of electricity prices. If the cable allows (does not sustain damage), the battery is charged or discharged to a pre-defined state of charge. When the battery is necessary to prevent or decrease cable damage, the battery tries to do so as much as possible. The strategy does not buy electricity on the APX market, thus only utilises the Imbalance Market for purchasing electricity.

	APXBidding				
ChargeDecision	**MIPGEET**	**DPGEET**	**RHGEET**	**Stochastic**	**None**
Memory-APXBidding	P1-MIPGEET	P1-DPGEET	H3		
CableGreedy	P2				H1
MIPGEET-small	P3				
PriceHeuristic			H4	H2	

Table 6.1: Strategies categorised by APXBidding and ChargeDecision procedures.

H2 (StochasticAPX, PriceHeuristic) combines two fast heuristics to form decisions on charge rates and APX bidding, Stochastic and PriceHeuristic. H2 can be seen as an advanced version of the H1 strategy, that incorporates thresholds in order to make a profit of electricity while greedily protecting the grid.

H3 (RHGEET, Memory-APXBidding) is an adapted version from the H_2 algorithm introduced in Ramezani et al. [3] and discussed in Section 4.2.3. The algorithm is used to decide on the volumes of the APX market. Since this algorithm does not deal with different electricity markets, the APX bid is always followed, an no electricity is purchased on the Imbalance Markets.

H4 (RHGEET, PriceHeuristic) combines the RHGEET heuristic with the PriceHeuristic (Figure 6.3), to see if this improves the solution over the H2 and H3 strategies.

Optimisation Strategies

Optimisation strategies make use of the DPGEET and MIPGEET algorithms from Chapter 5. These algorithms take some time to find an optimal solution. Strategies P1-DPGEET and P1-MIPGEET are devised to show the difference in running time and performance between the DPGEET and MIPGEET algorithms. As these schedules are based on forecasts, the charge rates may have to be adjusted later on. The P2 and P3 strategies have different ways of changing the original schedule.

P1-DPGEET (DPGEET, Memory-APXBidding) utilises the DPGEET algorithm for controlling the battery. The algorithm is run once per day, when the bids for the APX market are collected. The algorithm decides on the APX volumes. The P1-DPGEET strategy bids at a price such that the bid is guaranteed to be awarded (high price in case of purchasing electricity, low price in case of selling electricity). The strategy never changes its charging decisions. When the DPAGE algorithm is run, nearly all of the variables are forecasted, and can change later on. If at a later time the values have changed, the strategy does not change the initially proposed charge decisions.

P1-MIPGEET (MIPGEET, Memory-APXBidding) is identical to the P1-DPGEET strategy, except the MIPGEET algorithm is run instead of the DPGEET algorithm. This way a comparison in the running times and solution quality between the algorithms can be made.

P2 (MIPGEET, CableGreedy) is an advanced version of the P1-MIPGEET strategy. While the P1-MIPGEET strategy does not change values on intervals between APX bidding moments, the P2 strategy tries to resolve grid problems greedily. In case an unexpected grid issue occurs, the strategy adjust the previously decided charge or discharge rate accordingly.

P3 (MIPGEET, MIPGEET-small) also is an advanced version of the P1-MIPGEET strategy. At each interval, this strategy runs the MIPGEET algorithm. This algorithm can therefore make use of price peaks in the Imbalance Market. This strategy will demand more running time as the MIPGEET algorithm will be called upon at each interval, instead of once per day.

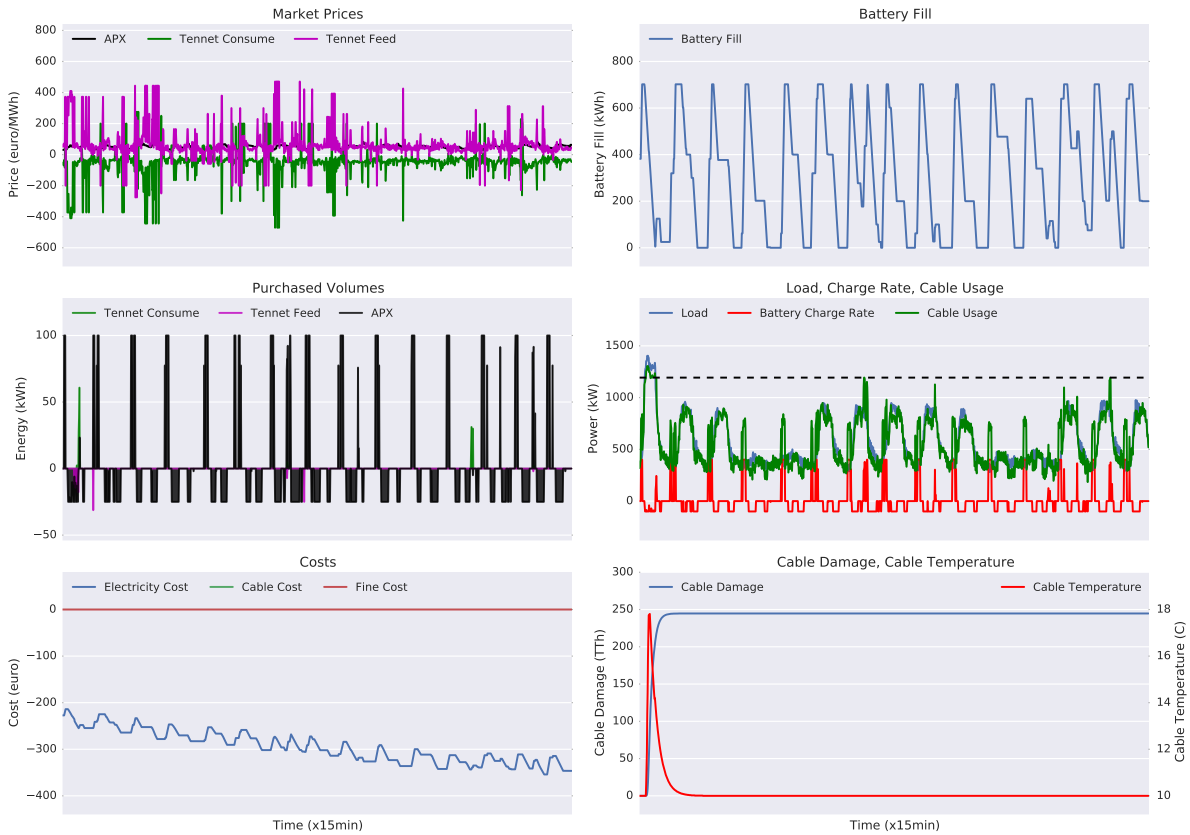


Figure 6.4: Example simulation results of the P3 strategy applied to Scenario B using the *realistic* forecaster (first repetition). For visualisations of all other strategies and scenarios see the online documentation page.

6.5 Results

The strategies from Section 6.4 were applied to each of the scenarios (Section 6.3) for 50 repetitions. The results of these experiments can be found on the online documentation page. At this page, for each scenario and strategy the first repetitions has also been visualised (Figure 6.4). Below we will discuss some of the interesting findings per scenario. The performances of the various strategies in terms of cost, cable damage and decision time are listed in Table 6.2. Costs, cable damage and decision times are compared in terms of mean values over the 50 repetitions. The bar charts visualise the standard deviations of the results over the repetitions.

6.5.1 Scenario A

Scenario A was constructed such that the cable maximum is higher than the maximum load demand (peak demand). Hence, not using a battery resulted in zero cable damage. This also means that badly operating the battery may result in unnecessary damage to the cable. In Figure 6.5 the cable damage is shown. In all but three cases cable damage occurred (H3 strategy and both P1 strategies, realistic forecaster). This shows that when using a realistic forecaster, a battery planning should continuously be reviewed. The advanced versions of P1, the P2 and P3 successfully adjust in order to prevent cable damage.

When we look at the total cost of electricity, we see big differences in cost between simulations using the realistic and perfect forecaster. For all optimisation algorithms, the profit using the perfect forecaster is

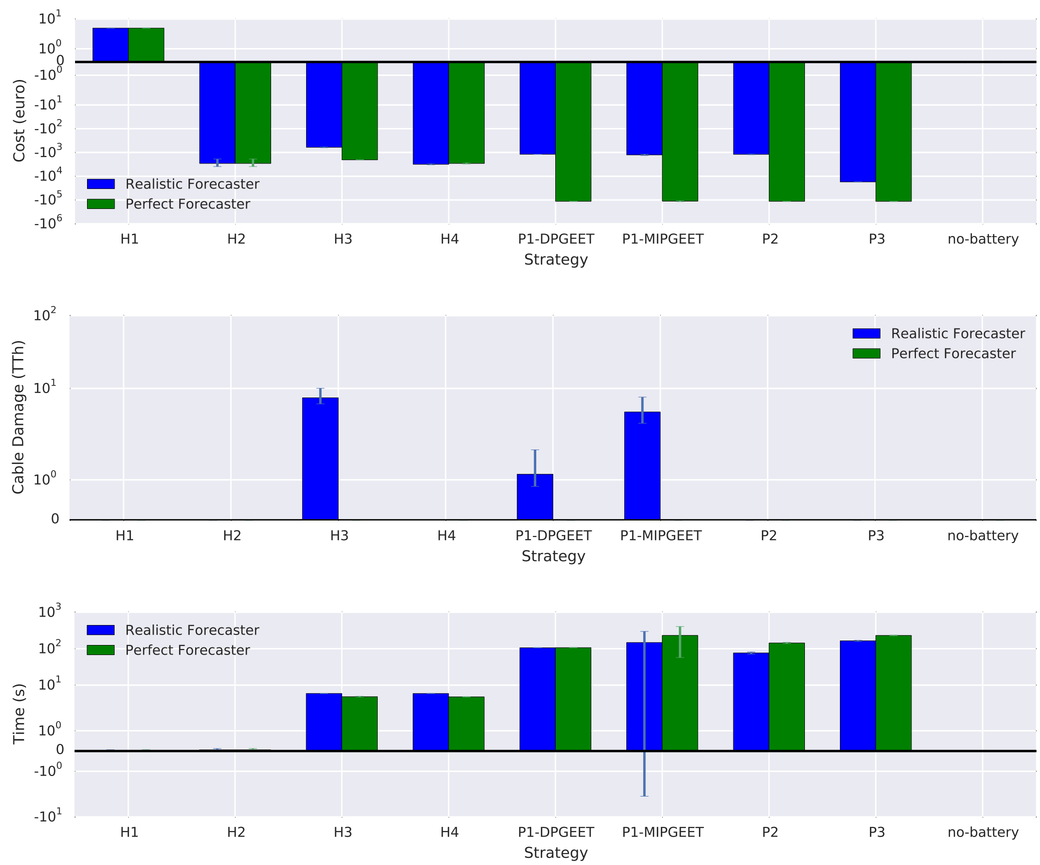


Figure 6.5: Simulation results from Scenario A. Top to bottom: Total Cost, Total Cable Damage and Total Decision Time. The error indicators show the variance over 50 repetitions. Note that the y-scale is a symmetric log scale.

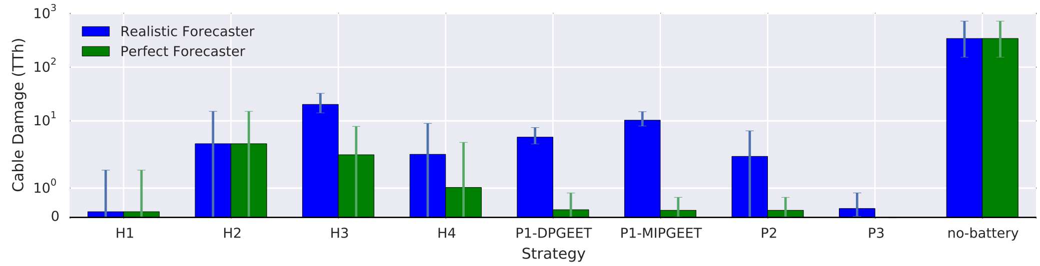


Figure 6.6: Cable Damage per strategy using adjusted Scenario A: $P^c_{\max} = 280$ kW. Note that the y-scale is a symmetric log scale.

over 100-fold the profit using the realistic forecaster. This is mainly due to the ability to use the imbalance market with prior knowledge. Unsurprisingly, the strategies that do not use a forecaster (H1,H2) have equal costs for both forecasters.

When using the realistic forecaster, the strategies that yields the most profit is the P3 strategy. In case the perfect forecaster is used, all optimisation strategies perform almost equally well (within 1% of each other). This result shows that optimisation strategies are not only preferred when an error-free forecaster is used, but also if we use the realistic forecaster that performs as well as literature suggests the current state of the art is. So in both cases, the P3 strategy is the preferred strategy.

The decision times in Figure 6.5 unsurprisingly show that the optimisation algorithms take much more times than the heuristic algorithms. The RHGEET algorithm (H3, H4) demands several iterations and therefore requires more time than then H1 and H2 strategy. The P1-MIPGEET strategy requires the longest running times, although the differences between the optimisation strategies (P1-DPGEET, P1-MIPGEET, P2, P3) were not significant ($p > 0.05$, paired-t-test).

Scenario A used a maximum cable capacity $P^c_{\max}$ that was higher than the maximum load demand over the entire simulation. Therefore, only three strategies (unnecessarily) caused cable damage. If we replace the cable capacity with a more critical capacity ($P^c_{\max} = 280$ kW), we observe cable damage in all but one case (Figure 6.6). optimisation strategy P3 succesfully prevents all cable damage when the perfect forecaster is used. In case the realistic forecaster is used, P3 also performs the best in preventing the cable damage. Given that the P3 strategy also resulted in the lowest electricity costs using the realistic forecaster, we believe the P3 strategy is superior in this scenario. The strategy takes around 200 seconds on our machine to make all decisions for a single year, equating to less than a second of running time per day. The algorithm therefore is quick enough for practical use.

6.5.2 Scenario B

In contrast to Scenario A, Scenario B has a cable with a maximum capacity lower than maximum load demands. Therefore, we see that in case no battery is used, significant cable damage occurs (Figure 6.7). Excluding the H1 strategy, for both the realistic and perfect forecaster the P2 strategy reduces cable damage the most. This means that the other optimisation strategy that adjusts its original planning, P3, produces more cable damage. Closer inspection of the data shows that the greedy approach of the P2 strategy results in short periods of high cable usage. The P3 algorithm spreads cable damage over the time intervals. As discussed in Höning et al. [5], short periods of high cable usage are economically preferred in terms of cable damage. This result confirms that our simulation model closely reflects real-world situations. The result also shows us that per-interval adjustments can be made greedily, removing the need for unwanted computational time. However, the P2 strategy results in significantly less profit in case the realistic forecaster is used (> 11 fold increase, $p < 10^{-26}$, paired t-test). The strategy that reduces cable damage the most, the simple H1

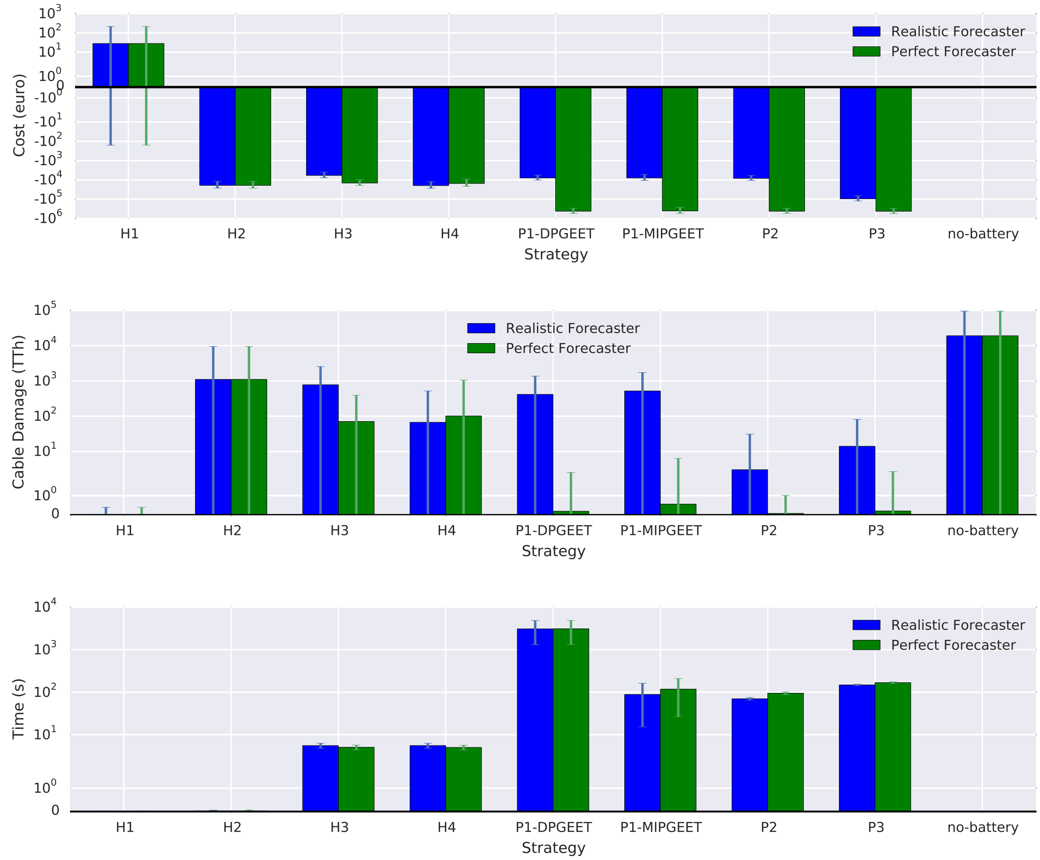


Figure 6.7: Simulation results from Scenario B. Top to bottom: Total Cost, Total Cable Damage and Total Decision Time. The error indicators show the variance over 50 repetitions. Note that the y-scale is a symmetric log scale.

strategy is quick and reliable, but doesn't attempt to create a profit by buying and selling electricity. For comparison, the average electricity cost of this strategy is €28.10 . The average cost of the P3 strategy is €-95,200.00. The H1 strategy is therefore good for grid enhancement, but will result in a lower profit and less profitable investment.

In terms of running times the strategies perform equally well compared to Scenario A, except the P1-DPGEET strategy (in relation to each other). This strategy takes over 34 times as long on average compared to the equivalent strategy that uses the MIPGEET algorithm ($p < 10^{-15}$). This is caused by the much higher battery capacity $C^B_{\max}$, which directly influences the state-space and subsequent running time of the DPGEET algorithm.

6.5.3 Scenario C

In Scenario C we evaluate the strategies on data that reflects storage of solar electricity at households. As explained in Section 6.3.3, no cable was modelled. Figure 6.8 shows how the P3 strategy yet again performs the best when the realistic forecaster is used. In case a perfect forecaster is available, all optimisation strategies perform equally well. The running time of the P1-DPGEET strategy, which was by far the largest in Scenario B, is now the smallest of the optimisation strategies. Again, the running time of this strategy is determined by the battery capacity. In this scenario the battery capacities are much lower, resulting in a

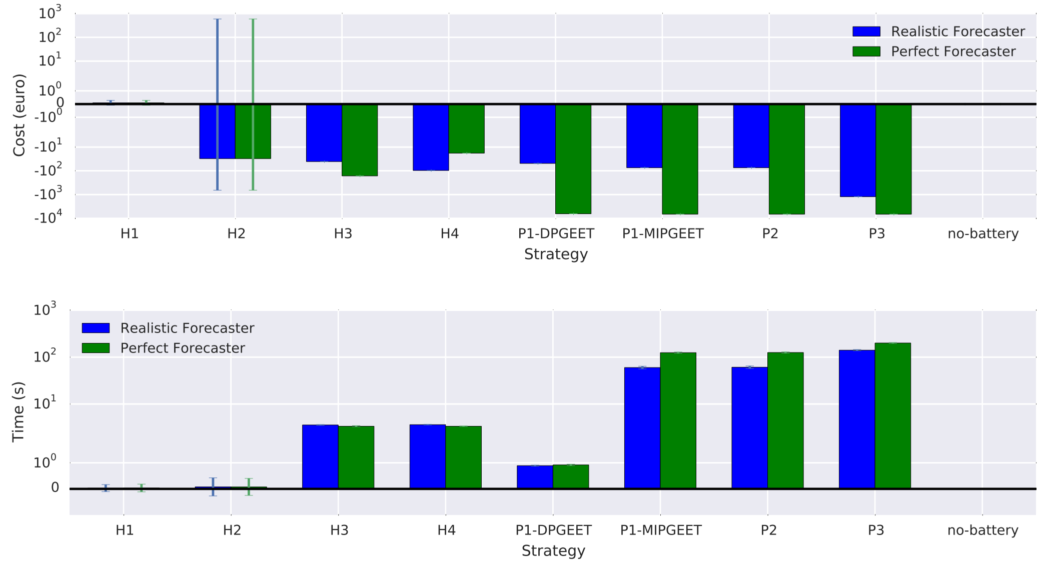


Figure 6.8: Simulation results from Scenario C. Total Cost (top) and Total Decision Time (bottom). The error indicators show the variance over 50 repetitions. Note that the y-scale is a symmetric log scale.

faster algorithm.

6.5.4 Performance under Varying Battery Properties

In the previous scenarios we evaluated the strategies on situations with different node loads, cable specifications and battery properties. In terms of influencing the eventual costs and cable damage, we speculate the the battery storage capacity and charge rate are crucial simulation parameters. For instance, "bigger" batteries are able to negate peak demands for longer periods of time. Furthermore, a bigger battery requires more energy to be fully charged, which results in different electricity costs. We therefore examined how well the strategies performed if we varied these parameters. The main question here is whether optimal strategies remain optimal under different battery properties.

Our main focus here is whether or not strategies are able to prevent more cable damage if we incorporate larger batteries in our simulation. This means we leave out Scenario C, since no cables were used in this scenario. We ran the simulations using the specifications as before (Section 6.3), as well as the simulations with 50% and 150% of the original battery capacity $C^B_{\max}$ and charge rates $P^B_{\max}$, $P^B_{\min}$. We chose to scale the battery capacity and charge rates linearly as both have an influence on the performance of the battery. A larger charge capacity allows a battery to endure longer peaks periods, while increasing the maximum charge and discharge rates allows for preventing damage from high power, but not necessarily long-time peaks. It is also reasonable to scale charge capacity and charge rates, as batteries consist of multiple sub-units. Increasing the size of the storage means increasing the number of sub-units. This increases both the allowed charge rates as well as the storage capacity.

In Figure 6.9 the new costs, cable damages and decision times of the strategies under varying battery properties are shown. In the previous analysis the P3 strategy consistently acquired the highest profit in terms of electricity cost. In this experiment this remains the same, regardless of the battery size or test scenario. We see that nearly all strategies have lower electricity cost, with the exception of the H1 strategy. This is to be expected, as the H1 strategy is the only strategy that does not include electricity prices in its decision protocol.

Interestingly, not all strategies prevent more cable damage if we specify a larger battery. The H3, P1-

scenario:	A			B			C		
	cost	dmg.	time	cost	dmg.	time	cost	dmg.	time
Realistic forecaster									
H1	*4*	0	0	*28*	**0**	0	*0*	0	0
H2	-2897	0	0	-19313	1105	0	-31	0	0
P1-DPGEET	-1192	1	106	-7949	415	*3094*	-49	0	1
P1-MIPGEET	-1267	5	147	-7963	520	89	-74	0	60
P2	-1202	0	76	-8314	3	70	-74	0	61
P3	**-17364**	0	*164*	**-95239**	14	149	**-1225**	0	*141*
H3	-607	*7*	6	-5885	777	6	-41	0	4
H4	-3156	0	6	-19741	67	6	-97	0	4
no-battery	0	0	**0**	0	*19114*	**0**	0	0	**0**
Perfect forecaster									
H1	*4*	0	0	*28*	**0**	0	*0*	0	0
H2	-2897	0	0	-19313	1105	0	-31	0	0
P1-DPGEET	-114164	0	107	-420683	0	*3108*	-6418	0	1
P1-MIPGEET	-112222	0	*231*	-406127	1	119	-6665	0	124
P2	-114301	0	143	-421214	0	95	-6665	0	125
P3	**-114645**	*0*	231	**-423105**	0	168	**-6691**	0	*200*
H3	-2045	0	5	-14582	71	5	-164	0	3
H4	-2898	0	5	-15387	101	5	-18	0	3
no-battery	0	0	**0**	0	*19114*	**0**	0	0	**0**

Table 6.2: Mean total cost (cost, in €), cable damage (dmg., in TTh) and total decision time (time, in seconds) from various strategies in the A, B and C scenarios using the realistic forecaster. **Bold** numbers indicate unique best performing strategies per category, *italic* numbers indicate unique worst performances.

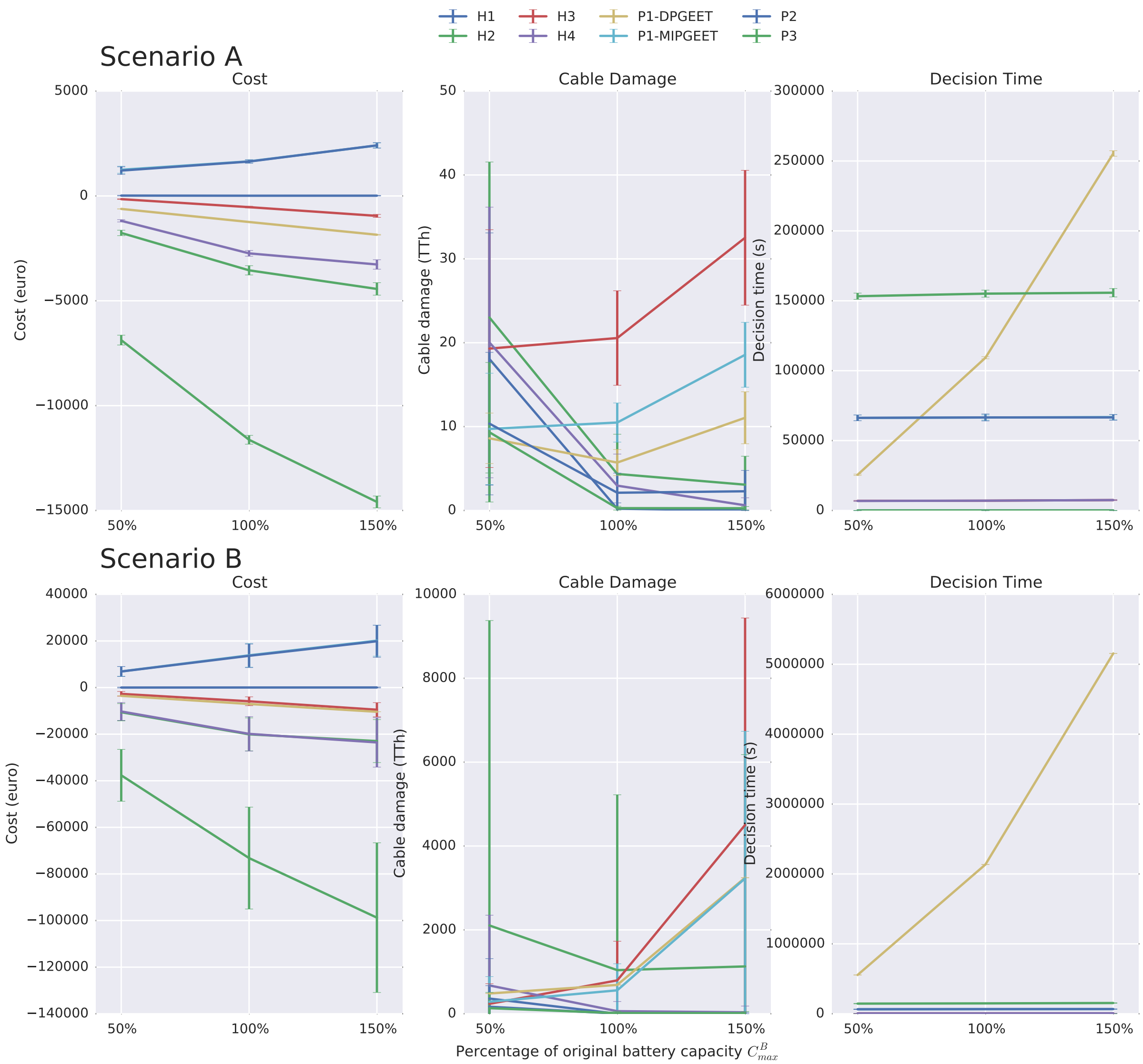


Figure 6.9: Electricity cost, cable damage and decision time of battery strategies under varying fractions of battery capacity ($C^B_{\max}$) and maximum charge/discharge rates ($P^B_{\max}$,$P^B_{\min}$) in comparison to the original A & B scenarios.

DPGEET and P1-MIPGEET strategies all perform worse in terms of battery storage. Manual inspection of the storage decisions shown that these strategies compose strategies that rely heavily on near-perfect decision making. Incorporating a buffer factor (reserving a fixed part of the storage capacity) would likely lead to improved damage prevention in these cases.

The P1-DPGEET strategy, which utilises the DPGEET algorithm, is the only strategy that requires longer computation time with larger battery capacity $C^B_{\max}$. As noted in Section F.1.2, this could be prevented by an alternative approach, possibly at the cost rounding errors that lead to higher costs or cable damage.

In conclusion, the altered battery properties don't lead to vastly different results in strategy performance. The P3 strategy remains the best-performing strategy in terms of electricity cost and is highly competitive in terms of preventing cable damage.

6.6 Discussion

In this chapter we presented a simulation model that evaluates strategies for energy storage in electricity grids. Using this simulation model we can test how strategies perform using different electricity grids, multiple energy markets and variable forecasting precision. The simulation model presented in this chapter also includes non-linear events such as cable breakage, although this function was not used in the evaluation for a lack of sufficient data.

The evaluation of the different strategies showed the importance of precise forecasting. In nearly all cases the perfect forecaster resulted in >100 times larger electricity profits. If we use the realistic forecaster, we see that the P3 strategy performs the best. This strategy combines a daily long-term and quarterly short-term planning to make decisions. This allows the running time to be sufficiently low for practical use and the utilisation of both forecasted and available data. The evaluations in this chapter have shown that in order to answer the question how profitable storage as grid enhancement can be, we must execute our calculations using the P3 strategy.

Chapter 7

Local Searching Strategic Parameters for Energy Storage

Contents

7.1 Introduction

In Chapter 6 we evaluated the performance of numerous strategies. The performance of each of these strategies are dependent on their parameters. For planning algorithms, these parameters included their planning length, i.e. the length of the horizon that battery schedules were made on. For other algorithms, these parameters determined when to charge or discharge the battery depending on the cable usage, electricity prices or state of charge.

In Chapter 6 we chose the parameters by hand. For instance, the planning length was chosen at 3 days. We argued that the planning length should be long enough to span at least 1 electricity peak (which typically occur once a day), but short enough to produce battery schedules within reasonable time. Some parameters were more difficult to estimate, as a set of parameters combined determined the performance of a strategy. For instance, the H2 strategy utilises 11 different parameters.

In this chapter, we present a systematic approach of determining these parameters by applying local search with random starts. Such an approach can be used to determine optimal parameter values for a specific situation. Since the method returns the optimal parameter values for historic data, we also asses how well these parameter values perform on a separate data set.

7.1.1 Local Search

Local search is a technique for approximating solutions to optimisation problems by exploring the neighbourhoods of solutions in solution space. The principle here is to slightly alter a solution $S \leftarrow S^*$ and to evaluate the quality of the new solution S^*. Many variants of local search exist, such as simulated annealing, stochastic hill climbing and gradient descent. Here, we apply continuous hill climbing with random starts. Note that *continuous* indicates that we explore continuous variables. For a fixed number of iterations, we choose a random value for each parameter. Then, we iteratively try to improve the solution by exploring solutions in which we alter each parameter separately. The pseudo-code for the continuous hill climbing procedure is provided below:

Algorithm 5 Generic Continuous Hill Climbing Algorithm(P_{init},S_{init},acc,ϵ)

```
P ← P_init
S ← S_init
C ← {−acc, −1/acc, 0, 1/acc, acc}
repeat
  COST_BEFORE ← evaluate(P)
  for i=1 to |P| do
    COST_BEST ← ∞
    for all c ∈ C do
      P_i ← P_i + S_i c
      if COST_BEST = ∞ or evaluate(P) < COST_BEST then
        c* ← c
      end if
      P_i ← P_i/(S_i c)
    end for
    P_i ← P_i + S_i c*
    S_i ← S_i c*
  end for
until evaluate(P) − COST_BEFORE < ϵ
```

The continuous hill climbing algorithm is invoked with an initial parameter setting P_{init}, an initial set of step sizes S_{init}, an accelerator value acc and a stopping criterion ϵ. The step size S_i, together with the accelerator value acc determine the new values for P_i which are explored. The algorithm uses 5 *candidates* to run for each parameter. The candidate set C uses the accelerator value acc to calculate these values ($\{-\text{acc}, \frac{-1}{\text{acc}}, 0, \frac{1}{\text{acc}}, \text{acc}\}$. The candidate with the lowest cost is set as the new parameter value (not that candidate 0 allows a parameter to remain unchanged). After each parameter was (potentially) improved, the new best solution is compared to the solution before optimising each parameter. In case the improvement is less than ϵ, the algorithm terminates.

Although more complex local search techniques exist, continuous hill climbing with random starts is a robust method for finding optimal solutions that will demonstrate whether local search can provide useful parameter settings for our strategies.

7.2 Method

From the strategies proposed in Chapter 6, we selected the H1, H2 and H4 strategies for parameter improvement. The H1 strategy is interesting as it has a single target parameter (τ, Section 6.4) which affects the strategy performance. The H2 strategy is the most heavily parameter-dependent strategy. Applying local search may potentially influence the output of this strategy the most, although there is a risk of over-fitting the parameters to a specific data set. The H4 strategy was selected as this strategy is an interesting cross-over between the parameter-heavy H2 strategy and a heuristic long-term strategy. As the local search algorithm runs the simulation for an extensive set of parameters, a fast planning strategy is useful here, which is why we selected the H4 strategy over the other planning strategies.

For each strategy, we specified an initial domain per parameter to randomly chose a value from. The set of parameters and initial domains are listed in Table 7.1. In each run, the local search algorithm generated a random parameter set, and iteratively improved the parameter setting using the continuous hill climbing algorithm. For each strategy we performed 100 runs.

The local search was executed using a simulation model using the Storage-Cable-Node model. The battery, node and cable were modelled identical to Scenario A (Section 6.3.1), except the cable maximum

Strategy	Parameter	Description	Initial domain
H1	τ	Battery target State of Charge (SoC)	$[0, 1]$
H2		APX bid price for charging	$[-1, 1]$
		APX bid volume for charging	$[0, 1000]$
		APX bid price for discharging	$[-1, 1]$
		APX bid volume for discharging	$[-1000, 0]$
		APX charge:discharge bid ratio	$[0, 1]$
	τ_{lower}	Battery lower segment threshold	$[0, 1]$
	τ_{upper}	Battery upper segment threshold	$[0, 1]$
	ρ^{+}_{lower}	Battery lower segment charge price	$[-1, 1]$
	ρ^{-}_{lower}	Battery lower segment discharge price	$[-1, 1]$
	ρ^{+}_{mid}	Battery mid segment charge price	$[-1, 1]$
	ρ^{+}_{mid}	Battery mid segment discharge price	$[-1, 1]$
	ρ^{+}_{upper}	Battery upper segment charge price	$[-1, 1]$
	ρ^{+}_{upper}	Battery upper segment discharge price	$[-1, 1]$
H4	T	Planning length	$[0, 500]$
	τ_{lower}	Battery lower segment threshold	$[0, 1]$
	τ_{upper}	Battery upper segment threshold	$[0, 1]$
	ρ^{+}_{lower}	Battery lower segment charge price	$[-1, 1]$
	ρ^{-}_{lower}	Battery lower segment discharge price	$[-1, 1]$
	ρ^{+}_{mid}	Battery mid segment charge price	$[-1, 1]$
	ρ^{+}_{mid}	Battery mid segment discharge price	$[-1, 1]$
	ρ^{+}_{upper}	Battery upper segment charge price	$[-1, 1]$
	ρ^{+}_{upper}	Battery upper segment discharge price	$[-1, 1]$

Table 7.1: Parameters and initial domains per strategy for local search optimisation.

capacity was set at $P^{c}_{\max} = 280$ kW, to create a more critical situation (in Scenario A, no cable damage occurred using most strategies). Market data on both the APX and TenneT Imbalance market was taken from 2013. The *difficult* forecaster was used to provide forecasts.

On the initial (randomly chosen) and final (after convergence) parameter set, we also ran the simulation model using market data from 2014 and a different sampling of *Zonnedael* load demand data. Where the 2013 simulation model was used a *training* set of the local search algorithm, the 2014 simulation was used as a *test*. We hypothesise that parameters that are closely fitted to the exact electricity prices and load demand will perform worse on the test simulation. Evaluating parameters on both a training and a test simulation allows us to analyse how transferable parameter settings are for each strategy.

7.3 Results

Here we analyse the performance of the three strategies while being optimised in the local search algorithm. The left column of Figure 7.1 shows the iterative convergence of each of the random starts per strategy. On the horizontal axis each point indicates the next generation of solutions, while the vertical axis indicates the performance of the solutions in the simulation. Of course, solutions can only improve in later generations, as only beneficial changes are excepted. Each line shows a separate random initial start configuration.

The figure shows that the random starts that need the most iterations to converge are not necessarily the random starts that ultimately lead to the lowest costs. For all strategies except H1, there are multiple local minima in which the algorithm converges. On the other hand, the H1 strategy ultimately finds solutions of the equal quality. Perhaps surprisingly, the optimal value for τ is 0.308, indicating that the battery charge level can be kept relatively low in this scenario. The H2 strategy has the largest range in costs for the

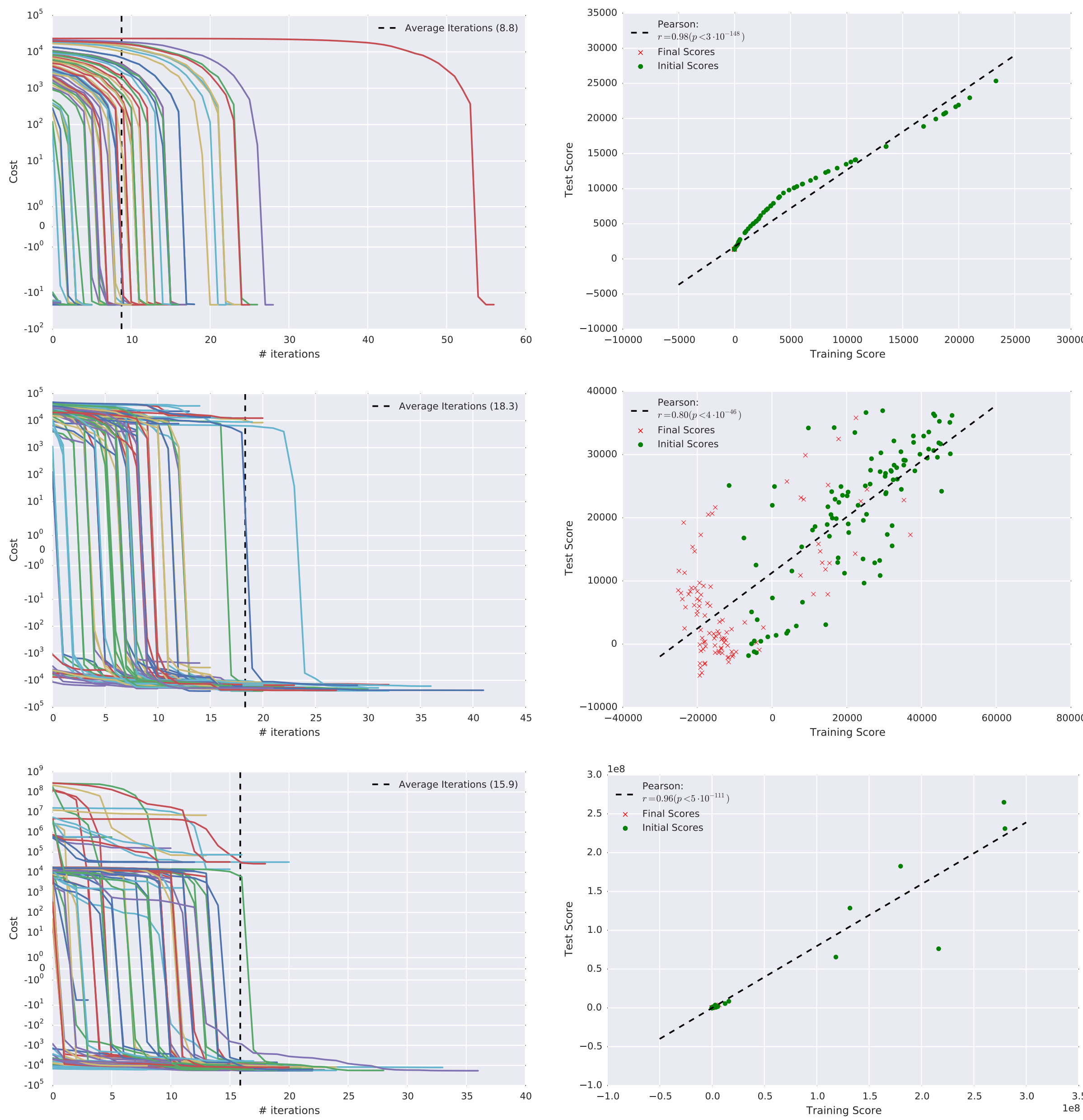


Figure 7.1: Iterative improvement of the hill-climbing algorithm with random starts a (left-column) and correlation between training and test set performance (right-performance) for various strategies. *(top to bottom)*: H1, H2 and H4 strategies

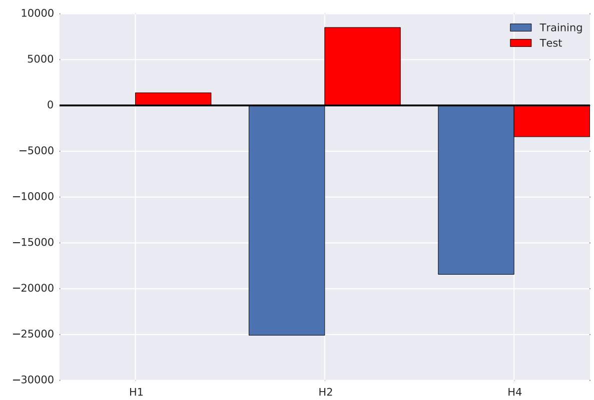


Figure 7.2: Best observed training score and corresponding test score per strategy.

initial parameter values, and ultimately also to the lowest final cost at -25076, reached after 15 hill-climbing iterations.

We performed an additional simulation on each of the best performing parameter sets per strategy, in order to evaluate how well the parameters performed in a related scenario. We therefore defined a *test* set using 2014 market data, and a different sampled version of the load demand. Figure 7.2 shows how well these optimal parameters settings performed on both the training and test simulation. As mentioned before, the lowest cost was observed for the H2 strategy at -25076. However, if we apply the same parameters to the *test* simulation, the H2 strategy performs much worse than the other strategies. This is likely due to a change in overall electricity prices between 2013 and 2014. In 2013, the average electricity price on the APX market was 51.93 €/MWh ($\sigma = 13.5$). In the following year the electricity prices went down significantly, to an average price of 41.18 €/MWh ($\sigma = 10.69$). When deciding to charge or discharge, the H2 strategy relies heavily on its parameters. The discrepancy between *training* and *test* costs further emphasis this dependence. In contrast to the large difference observed for the H2 strategy, the H4 strategy performance decreases less in the training simulation. This implies that the chosen parameters are more transferable between years as compared to the H2 parameters. Intuitively, this makes sense if we look at some of the parameters. For instance, the optimal value for the planning length T of H4 is 481 intervals, slightly more than five days. Although the average electricity prices might change over time, this planning length is a property that indicates the usefulness of looking far-ahead in time. As this planning length is more than 5 days, it implicates that given the price patterns, forecasting precision and load demand five days is far enough to find optimal solutions, while being flexible enough to correct unexpected changes.

The claim that some parameters settings are more applicable on different scenarios can be further examined is we calculate the correlation between *training* and *test* scores of the initial and final parameter settings (Figure 7.1, right-column). Indeed, we find the lowest correlation ($r = 0.80$) between training and test scores for the H2 strategy. The highest correlation ($r = 0.98$) is observed for the H1 strategy, although this correlation may be heavily influenced by large-valued initial scores. For all strategies, significant correlations are observed, indicating the in all cases the parameter settings that perform well in the training simulation are expected to perform well in the test simulations as well. The correlation between test and training performance for the H4 strategy is also relatively high ($r = 0.96$). It seems that the use of strategy that makes a planning (even if it uses a heuristic) makes the parameter setting more robust for changes in the data.

7.4 Discussion

We have shown that we can successfully find parameter settings that perform well in not only simulation that were used to execute local searches on, but also on separate simulations. This indicates that important parameters such as the planning length T and battery target τ are transferable between different situations. When batteries will be implemented it's therefore not necessary that data of the exact situation is known beforehand. However, as the convergence on the training simulations have shown, precise-parameter modelling can decrease the cost of operating a battery in such situations.

Local search is an approximation technique. As the optimal values of the parameters in this chapter are not known, we cannot evaluate the performance of the algorithms in terms of the global optima. Although the data was not formally presented here, the optimal values calculated by the local search algorithms have outperformed our manual estimates in all cases. The method presented in this chapter does still rely on the manual specification of the intervals from which random starts are sampled. This specification, and the evaluation of the eventual parameter-settings will still be manual processes. However, this technique does provide a systematic approach for a difficult task.

Chapter 8

Exploring Profitability of Energy Storage

Contents

In Chapter 5, 6 and 7 we designed, evaluated and optimised strategies for controlling batteries in electricity grids. In this chapter, we attempt to relate the performance of these strategies to real-world cases. The objective here is not to provide a catch-all answer to the question whether or not storage is economically profitable, but what the main factors contributing to this profitability are. In Section 8.1 we characterise the influence of forecasting precision on the performance of our strategies. This allows us to evaluate both the potential and current state of forecasting models. In Section 8.2 we show that our simulation model can be extended to incorporate a more complex grid structure, opening possibilities for more complex grid balancing objectives. Lastly, Section 8.3 provides a method to calculate the economical benefit of implementing storage in specific grid situations.

8.1 Planning using Forecasts & the Need for Precise Models

In Chapter 6 two levels of forecasting precision were used to evaluate the performance of various control strategies. In each scenario, there was a large difference in the profit made from energy trading between the realistic and perfect forecaster. Here we evaluate how increasing the forecasting precision leads to higher profit and less cable damage. Our comparison includes the H2 and P3 strategies using the realistic and perfect forecaster. We show how increasing the forecasting precision leads to higher profits and less cable damage. As a test case, we used the same simulation settings as the training case from Chapter 7.

In Figure 8.1 we show how the P3 strategy performs using different levels of forecasting precision, characterised by the Mean Absolute Scaled Error (MASE) value. See Section 6.2.5 for an explanation on how these forecasts were generated. Figure 8.1 also shows how the H2 strategy performs in this simulation, using the optimal parameter setting found using the local search procedure from the last chapter. Lastly, the figure

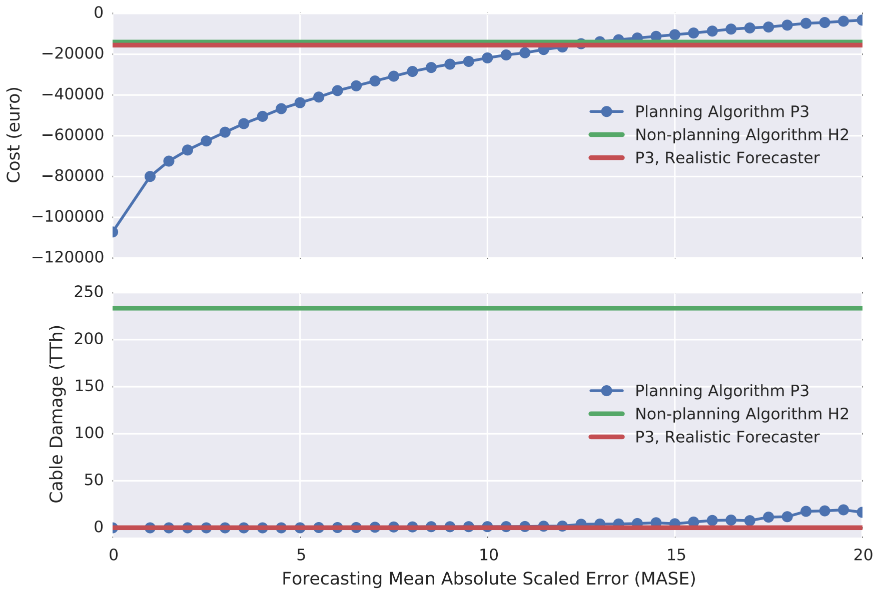


Figure 8.1: The Value of Forecasting: Electricity cost (top) and cable damage (bottom) using the P3 strategy as a result of varying forecasting precisions, compared to the performance of the H2 strategy and P3 strategy using realistic forecaster.

includes the electricity cost and cable damage when the battery is controlled using the P3 strategy with the realistic forecaster from Chapter 6.

Unsurprisingly, a lower MASE leads to higher profits (lower electricity costs) and less cable damage when the battery is controlled by the P3 strategy. When the precision increases (MASE decreases), the *rate* in which profits increase also increases. This shows that developing better forecast models will continue to be rewarding, as long as the precision is even slightly imperfect. In agreement with the evaluations in Chapter 6, the analysis shows that there still is a large difference in profit made using a perfect and realistic forecaster.

Another point that can be taken from Figure 8.1 is that while the profit generated by the heuristic strategy H2 is comparable to P3, in terms of cable damage P3 is superior. Furthermore, the cable damage prevention using the realistic forecaster is already at the level of perfect forecasting.

Perhaps unsurprisingly, Figure 8.1 shows that efforts in forecasting models will increase the profits from energy trading significantly. As the actual forecasting of electricity prices and demands is beyond the scope of this study, we can only guess what the practical bounds of the forecasting precisions are. Our model can however be used as a tool to evaluate how a forecasting model and subsequent improvements affect the profitability of energy storage.

8.2 Advanced Grid Structures

In Chapter 6 the performance of strategies and algorithms for battery control were evaluated on the Storage-Cable-Node model grid structure. In practice, electricity distribution grids obviously have more complex structures. Here we discuss three types of structures, and how well our model can be extended to incorporate these structures.

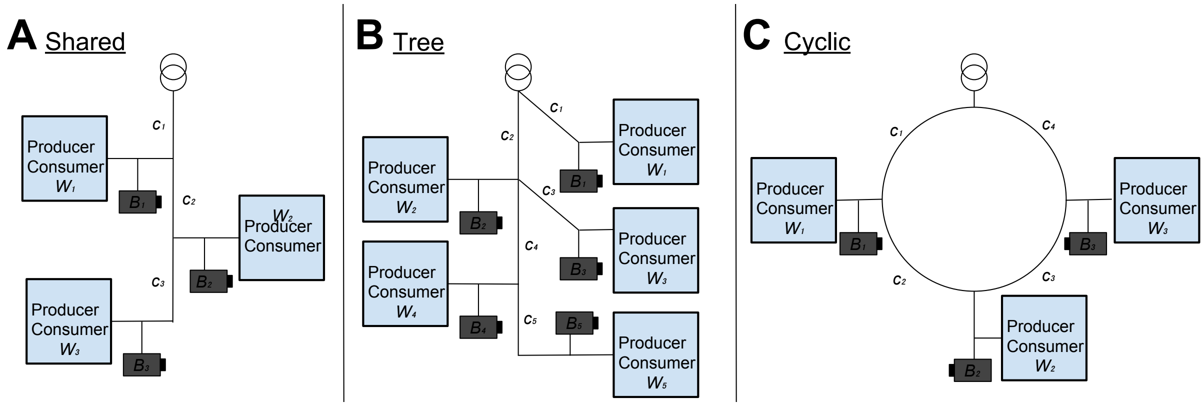


Figure 8.2: Three types of advanced grid structures.

8.2.1 Shared Cable Network

We first look at the *shared* cable network. In this network, a single cable c is modelled with multiple batteries attached. Consider the example network with three nodes and batteries in Figure 8.2A. In order to prevent cable damage, each cable segment (c_1, c_2, c_3) needs to endure less power than the set cable maximum $P^c_{\max}$. This creates different constraints on the system, as downstream cable segments are affected by less nodes and batteries. For instance, the last cable segment c_3 only sustains the sum of electricity currents from B_3 and W_3. Segment c_2 also connect B_2 and W_2 to the grid, and therefore sustains the sum of the demand by B_2,W_2,B_3 and W_3. One further step upstream, c_1 endures the sum of all batteries and nodes in this system. The cable capacity constrains therefore become:

$$-P^c_{\max} \leq P^{W_3}_t + P^{B_3}_t + P^{W_2}_t + P^{B_2}_t + P^{W_1}_t + P^{B_1}_t \leq P^c_{\max}, \quad 0 \leq t \leq T \tag{8.1}$$
$$-P^c_{\max} \leq P^{W_3}_t + P^{B_3}_t + P^{W_2}_t + P^{B_2}_t \leq P^c_{\max}, \quad 0 \leq t \leq T \tag{8.2}$$
$$-P^c_{\max} \leq P^{W_3}_t + P^{B_3}_t \leq P^c_{\max}, \quad 0 \leq t \leq T \tag{8.3}$$

These constraints can easily be incorporated in the Mixed Integer Programming algorithm from Section 5.4. We define n as the number of battery-node combinations on the network. We can achieve control for shared cable networks by replacing constraint (F.3) by:

$$-P^c_{\max} \leq \sum_{k=i}^{n} \left[P^{W_k}_t + P^{B_k}_t \right] \leq P^c_{\max}, \qquad 1 \leq i \leq n, 0 \leq t \leq T \tag{8.4}$$

A full formulation of this new algorithm is provided in Section F.3.3. To provide a demonstration of this algorithm we used the example network from above and data from different repetitions of Scenario A (see Chapter 6). Supplementary Figure C.1 shows battery charge states, charge rates and overall costs of a part of this simulation. The cable segments are shown from top to bottom. The middle column shows the charge levels of the three batteries. We see that the sustained demand is the largest in segment c_1. However, using the three batteries the algorithm effectively minimises the cable damage in each segment. Interestingly, this demonstrates that multiple batteries can work together to relieve grid issues. Furthermore, the necessary adjustment to the linear program formulation is minimal.

8.2.2 Tree Grid Structure

A more complex form of electricity grids are networks that are *trees*. In computer science, a tree is a structure with a root node or element that has other elements or trees as substructures. These elements or trees are defined as *children* of the root. Any element in a tree can have multiple children, while all elements (except for the root element) have one *parent* element. In Figure 8.2B an example of a tree structure is shown. The root element here is the connection to the external grid, depicted by the two overlapping circles. All nodes and batteries are children of this root. The tree structure implies that for each node or battery, there is a single path towards the external grid. Thus, the power necessary to supply the node or battery has to be transferred through all the upstream cables leading towards the external grid. For instance, in Figure 8.2B power to W_4 and B_4 passes through c_2 and c_4, while power to W_1 and B_1 only passes through c_1. In similar fashion to the shared cable structure in the previous section, we can replace equation (F.3) from the MIPGEET algorithm (Section 5.4) by a new set of equations:

$$-P_{\max}^{c_1} \leq P_t^{W_1} + P_t^{B_1} \leq P_{\max}^{c_1}, \quad 0 \leq t \leq T \tag{8.5}$$
$$-P_{\max}^{c_2} \leq P_t^{W_5} + P_t^{B_5} + P_t^{W_4} + P_t^{B_4} + P_t^{W_3} + P_t^{B_3} + P_t^{W_2} + P_t^{B_2} \leq P_{\max}^{c_2}, \quad 0 \leq t \leq T \tag{8.6}$$
$$-P_{\max}^{c_3} \leq P_t^{W_3} + P_t^{B_3} \leq P_{\max}^{c_3}, \quad 0 \leq t \leq T \tag{8.7}$$
$$-P_{\max}^{c_4} \leq P_t^{W_5} + P_t^{B_5} + P_t^{W_4} + P_t^{B_4} \leq P_{\max}^{c_4}, \quad 0 \leq t \leq T \tag{8.8}$$
$$-P_{\max}^{c_5} \leq P_t^{W_5} + P_t^{B_5} \leq P_{\max}^{c_5}, \quad 0 \leq t \leq T \tag{8.9}$$

Here, the grid structures defines for each cable segment c_i the downstream connected nodes and batteries. The sum of their powers should satisfy the maximum cable capacity constraint. Thus, by replacing the capacity constraints in the MIP formulation we can also accommodate grid structures that are trees.

8.2.3 Cyclic Grid Structure

In a tree structure, all elements (except the root) have a single *parent* element. If we remove this rule from our grids, we effectively allow all kinds of *graphs*. This may give rise to cycles: a path from element to element along 2 or more connections which returns at the start element. In Figure 8.2C an example of a cyclical electricity grid is shown. In a cyclical electricity grid it is much harder to determine the amount of power, voltage, current and resistance that occurs along the cables, nodes and batteries. This problem of finding these values is called the *load-flow calculation* of a grid.

In the model we've used in this thesis the assumption was made that the voltage was equal everywhere in the model. This allowed us to focus on the power P in our algorithms and strategies. In order to accommodate calculations on cyclical grids this assumption needs to be removed, and the voltage, current and resistance along all parts of the grid need to be considered. In van den Akker et al. [48] an approach to such modelling for algorithmic purposes is shown. The authors make use of Kirchoff's current law and Kirchoff's voltage law to formulate the dependencies of current and voltage along electricity cables, nodes and storage units. Using an elegant Gauss-Jordan elimination their method allows for incorporating cyclical grid substructures in their approach to establishing optimal storage locations [49].

8.2.4 Future Work

8.3 The Value of Storage in Electricity Grids

Here, we attempt to summarise and integrate the different benefits of storage in electricity grids in a single calculation. We will focus on the following ways storage can be of value:

Revenue A large part of this thesis has been dedicated to energy trading on both the APX and TenneT Imbalance market. A large storage asset with access to these markets can generate profit by charging the battery with electricity at a lower price than the price it is discharged with.

Grid Enhancement If the battery is located next to a bottleneck (power cable, transformer, etc.) in the electricity grid, the storage unit can prevent damage to the asset. Every asset in the electricity grid is installed for a pre-specified amount of time. Traditionally, when a cable or transformer is damaged by the high electricity flow in the network, this asset will be replaced by one that can sustain the higher electricity flows. A battery can postpone this reinforcement until the moment the asset would've been replaced under normal conditions. In the Netherlands, power cables are installed for 40 to 70 years. If after for example 20 years the cable would need replacement because of higher electricity use, 20 to 50 years of cable value would've been lost. However, if a battery can postpone this moment to the original, pre-specified replacement time, a system operator does not lose its cable value. This is the second mode of possible benefits by storage, grid enhancement.

Island operation In case of a failure in the electricity grid As discussed in the previous section, the battery can supply a node with electricity in case a cable in the grid fails, resulting in an isolation of the node. Island operation has never been the objective in this study so far. We believe that if island operation is simple, a fixed amount of battery capacity should be reserved at all times to take over the demand in the grid in case of a failure. This battery capacity should not be used by the control strategy, and thus is irrelevant for our earlier evaluations. However, a slight benefit can still achieved by using whatever charge level is present in the battery at the time of a failure.

8.3.1 Profitability of Storage in the Ettenleur SSU case

To asses the value generated by revenue, grid enhancement and island operation requires the use of our simulation model. The outcome of running the model relies on the establishment of grid, cable and battery parameters. Below, we provide an example calculation for the Ettenleur SSU case. The use of a real-world case provides parameters that can directly be used. The example calculation thus provides a usable estimate in the Ettenleur case, but also provides a method for calculating the value of storage in other situations where revenue, grid enhancement and island operation are beneficial.

In the Ettenleur SSU case, the technical capacities of the battery are specified. The battery has a maximum capacity of $C^B_{\max} = 230kW$, a charge rate interval $[P^B_{\min}, P^B_{\max}] = [-100kW, 400kW]$ and a round-trip efficiency $\mathcal{E} = 0.80$. The cable that is located near the battery has a maximum capacity ($P^c_{\max}$) that is higher than the maximum demand of the connected neighbourhood. This would negate the possible benefits of grid enhancement, as there is no cable damage to prevent. We will therefore examine the value of storage for different values of α, defined as the maximum demand from node W as fraction of the cable maximum capacity:

$$\alpha = \frac{P^c_{\max}}{\max\limits_{1 \leq t \leq T} P^W_t} \tag{8.10}$$

The lower α, the more critical the grid situation concerning cable c. As α represents the severity of the grid situation, it also implicates in which situations storage can be an alternative of traditional grid reinforcements.

The next step is to quantify the three modes of benefits. Revenue is straightforwardly expressed in money. Grid Enhancement can be expressed in terms of investment that is saved from the original grid reinforcement. The third mode, island operation is harder to quantify. We define *saveable* time to measure how long the battery can satisfy the demand of the node in case cable c fails. By running the simulation model and measuring the saveable time at each interval, we find a mean saveable time. Of course, saveable time is not an objective of the battery. But this measure does provide an indication of the additional benefit of island operation. Figure 8.3 shows the distribution of saveable time (left panel) under different values of α. The right panel shows the battery level and saveable time during a fragment of the simulations. In case a grid failure occurs and the DSO cannot connect a node to the grid, a fine has to be paid as compensation.

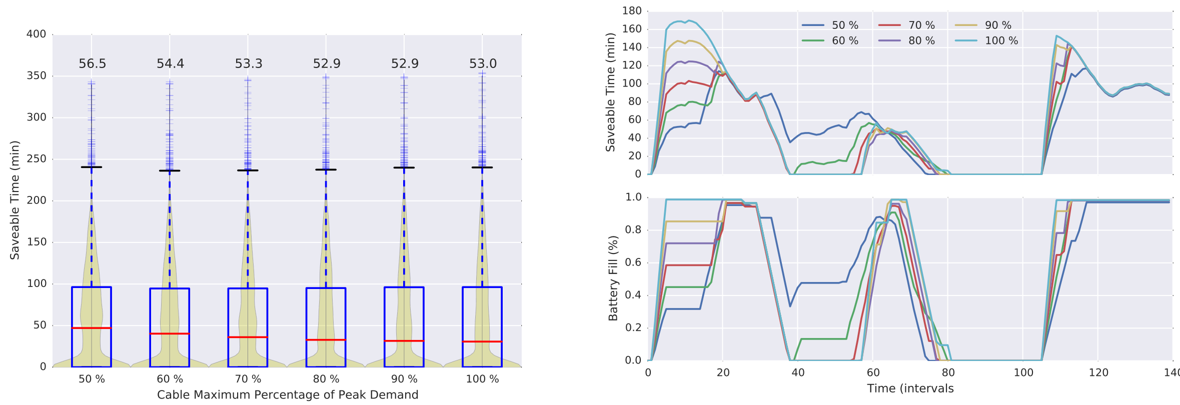


Figure 8.3: Saveable Time under different values of α. Left: distribution of saveable times over a complete simulation (numbers indicate the mean saveable time). Right: fragment of the simulation showing saveable time as a result of battery level for different values of α.

For households, this fine is €5 per hour. Multiplying the saveable time, number of households and this fine provides the island operation benefit of a battery in case of a cable failure.

We now introduce a formula which quantifies the benefits of battery B as function of α:

$$\text{yearly benefit}(\alpha) = f(\alpha) + \frac{\beta}{N} + h(\alpha)\gamma p_{\text{break}} \tag{8.11}$$

Formula (8.11) consists of three parts, that resemble the revenue, grid enhancement and island operation benefits storage presents. $f(\alpha)$ is the revenue generated with varying electricity prices, which account directly to the yearly benefits. For grid enhancement, postponing traditional grid reinforcements can only be of value if a situation occurs where the cable is being damaged without intervention, and adding a battery would prevent sufficient cable damage. Here, we determine the critical amount of yearly cable damage at 1000 TTh. If a battery can reduce the cable damage to less than 10 TTh, the yearly benefit of postponing a cable reinforcement is the cable cost β divided by the original cable lifetime N. The island operation is quantified by multiplying the mean saveable time in isolation $h(\alpha)$, the hourly cost of not-connecting W to the grid γ and the probability of a cable breakage per year p_{break}.

In the Ettenleur case, we consider a cable that is utilised for $N = 50$ years, as is common in electricity grids [6]. According to ter Haar [50], the cost of power cables in the distribution grid can either be €46, €63 or €110 per meter, depending on the type. If we consider protecting power cables between 500 and 5000 meter, this creates a range for β of €23,000 and €220,000. The yearly benefit of grid enhancement $\frac{\beta}{N}$ thus ranges from €$\frac{23000}{50} =$ €460 to €$\frac{550000}{50} =$ €$11,000$. For island operation we consider the probability of a cable breakage $p_{\text{break}} = 0.2$. The hourly fine for not-connecting W (a neighbourhood of 240 households)to the grid, γ, is $240 \cdot$ €$5 =$ €1200.

Now, we can evaluate the benefits of storage in electricity grids as function of α. In Figure 8.4 we summarise the values of $f(\alpha)$, $h(\alpha)$ and the grid damages that occur for different values of α. The cable damage that occurs with and without storage (Figure 8.4, bottom row) implies that only if $\alpha = 0.7$, a storage unit effectively prevents a previously critical situation. For lower values of α the situation is worse, but the battery cannot relieve the cable damage sufficiently. For higher values of α we find that there is not enough cable damage in the original situation that a storage unit is necessary. At $\alpha = 0.7$, $f(\alpha)$ averaged over 2012, 2013 and 2014 equates to €11,191. The saveable time on average is 0.95. Using these parameters (and the most beneficial scenario for grid enhancement), when $\alpha = 0.7$ the yearly profit equates to €11,191 + €11,000 + €232 = €22,423.

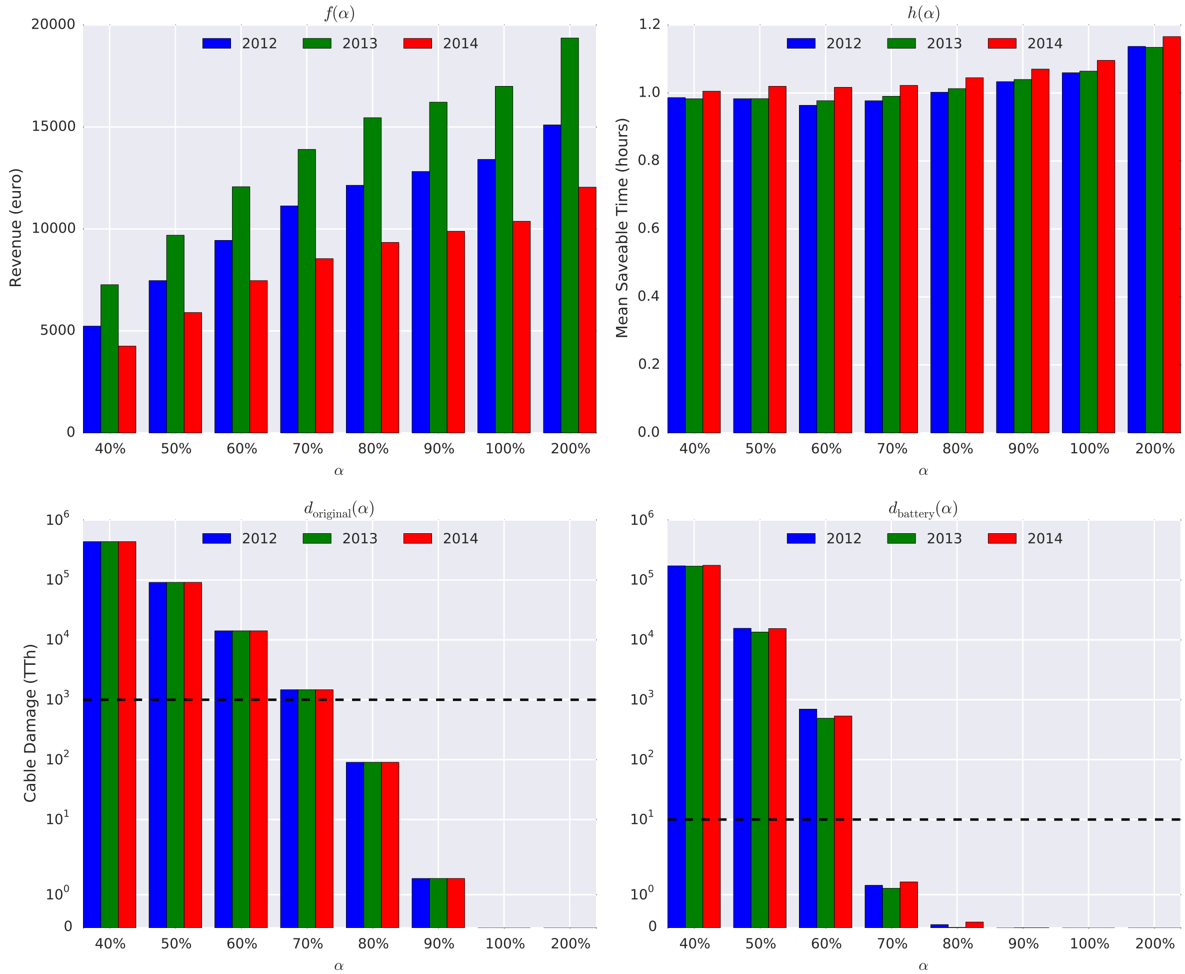


Figure 8.4: Revenue, saveable time and cable damage as result of different levels of grid criticality α. Top left: Revenue $f(\alpha)$, top right: Saveable time $h(\alpha)$, bottom left: cable damage without battery $d_{\text{original}}(\alpha)$, bottom right: cable damage with battery $d_{\text{battery}}(\alpha)$

The main question now becomes how this yearly benefit of €22,423 compares to the investment cost in batteries. Seba [51] provides an estimate that Lithium Ion batteries are available for \$600 per kWh (including electronics). This equates to a total cost of \$138,000 dollar for the Ettenleur SSU, which relates to roughly €125,000 using present conversion rates. If the battery can sustain usage for over 6 years, such an investment could prove beneficial. We admit that this conclusion contains a lot of speculation. First, we have not focused on how our algorithms degrade battery quality. Secondly, the cost and development of batteries is not thoroughly researched in this study. But the numbers do provide the notion that currently battery storage is very expensive, and that the prices of storage need to drop before grid enhancement becomes a viable option for grid enhancement.

Chapter 9

Discussion

In this study we've applied computer science practices to the electricity sector. We've analysed how electricity grids are increasingly experiencing peak demands, and how energy storage can relieve these grids. We've presented a simulation model that can be used to test strategies that control batteries for grid enhancement and energy trading. The simulation model is easily adaptable to different electricity grids, cable and battery models and market structures. This way, we were able to evaluate storage strategies using realistic models for electricity demand, electricity markets, batteries and power cables. As part of this simulation model, we for instance presented a parameter-based temperature model for cable damage. Additionally, the simulation model incorporates forecasting errors in its decision making protocol. We've also shown that local search can be used in combination with the simulation model to find useful decision parameters for the strategies.

The simulation model serves as a decision support tool that can test the performance of strategies and indicate the effectiveness and profitability of storage in a particular grid situation.

Still, whether or not storage will be a valuable addition to electricity grids in the future remains to be seen. Although our study has shown that using storage significant profits can be made on wholesale electricity markets, the value of storage in electricity grids in the future depends on numerous factors. Under current battery prices, the profit that can be made on the electricity markets is not yet sufficient to overcome the cost of batteries. These battery prices are expected to drop [51], but other factors are important as well. Here, we provide a brief summary of these factors.

There is an ongoing trend that European electricity grids and markets are becoming more and more connected. The precise impact these connections will have on price volatility and electricity availability are not yet known. On the other hand, the continuous increase of renewable energy (RE) production will create for larger differences in electricity prices between hours or even shorter periods. Price volatility (price differences) are essential for energy trading using storage.

Furthermore, the connection to European markets and a larger fraction of RE will affect the forecasting precision of electricity prices. Our efforts in Section 8.1 have shown that exact forecasting is not essential for generating profits using energy trading, but does heavily influences the profit of energy trading.

Another important factor in electricity grids will be the rate in which electric vehicles (EV) and other high-powered devices will be used in the future. Although the number of EVs in the Netherlands is small, the growth is stable and increasing [52]. More importantly though, is to what extent DSOs will account for the peak demands when they reinforce grid infrastructures. In the Netherlands, a lot of infrastructure was created during the economical growth of the sixties and seventies. As most power cables are used for around 50 years, a massive replacement operation of the Dutch electricity grid is due in the coming decades [6].

This replacement operation also means that the opportunities for economically profitable storage implementations are limited. In the previous chapter we've shown that postponing grid reinforcements greatly accounts to the investment reward of storage. The timing of implementing storage in grids relative to the moment of cable placement is therefore important as well.

The energy sector is rapidly changing. Continued renewable energy implementation, connected European electricity grids and markets, the increase of electricity consumption, all-electric households and other high-powered appliances put pressure on electricity grids. The future will show whether traditional grid enforcements will suffice in keeping the grid reliable and affordable. If not, energy storage presents a valuable way for grid enhancement. We've shown that there is a significant profit that can be made on electricity markets while maintaining electricity grids, but that under current battery prices few investment opportunities exist. The contributions of this thesis can be used in the future to decide if and when storage for grid enhancement is the preferred mode of operation.

Chapter 10

Bibliography

[1] Amotz Bar-Noy, Matthew P Johnson, and Ou Liu. Peak shaving through resource buffering. In *Approximation and Online Algorithms*, pages 147–159. Springer, 2009.

[2] Stefan Nykamp, Vincent Bakker, Albert Molderink, Johann L Hurink, and Gerard JM Smit. Break-even analysis for the storage of pv in power distribution grids. *International Journal of Energy Research*, 38 (9):1112–1128, 2014.

[3] Sara Ramezani, N Honing, and Han La Poutré. Fast and revenue-oriented protection of radial lv cables with smart battery operation. In *Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on*, pages 107–114. IEEE, 2013.

[4] Stefan Nykamp, Maurice GC Bosman, Albert Molderink, Johann L Hurink, and Gerard JM Smit. Value of storage in distribution grids - competition or cooperation of stakeholders? *Smart Grid, IEEE Transactions on*, 4(3):1361–1370, 2013.

[5] Nicolas Höning, Erik De Jong, Gabriël Bloemhof, Han La Poutré, and Ö Usta. Thermal behaviour of low voltage cables in smart grid–related environments. *Proceedings. of The 5th IEEE PES Innovative Smart Grid Technologies (ISGT 2014) European Conference*, 2014.

[6] Peter van Oirsouw and JFG Cobben. *Netten voor distributie van elektriciteit*. Phase to Phase, 2011.

[7] Erik van der Hoofd Bert den Ouden, Prof. Albert Moser. Market review 2014 h1. Technical report, Tennet, 2014.

[8] Heiko Hahn, Silja Meyer-Nieberg, and Stefan Pickl. Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3):902–907, 2009.

[9] Eva Gonzalez-Romera, Miguel A Jaramillo-Moran, and Diego Carmona-Fernandez. Monthly electric energy demand forecasting based on trend extraction. *Power Systems, IEEE Transactions on*, 21(4): 1946–1953, 2006.

[10] Javier Campillo, Fredrik Wallin, Daniel Torstensson, and Iana Vassileva. Energy demand model design for forecasting electricity consumption and simulating demand response scenarios in sweden. 2012.

[11] Hesham K Alfares and Mohammad Nazeeruddin. Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, 2002.

[12] Peter Nijhuis. *Electricity demand forecasting at low voltage level for smart charging electric vehicles*. PhD thesis, Eindhoven University of Technology, 2013.

[13] Ballard Asare-Bediako, WL Kling, and PF Ribeiro. Day-ahead residential load forecasting with artificial neural networks using smart meter data. In *PowerTech (POWERTECH), 2013 IEEE Grenoble*, pages 1–6. IEEE, 2013.

[14] P. Kadurek, W.L. Kling, P.F. Ribeiro, and J.F.G. Cobben. Electricity demand characterization for analyzing residential lv distribution networks. In *PowerTech (POWERTECH), 2013 IEEE Grenoble*, pages 1–5, June 2013. doi: 10.1109/PTC.2013.6652168.

[15] Xin Lu, Zhao Yang Dong, and Xue Li. Electricity market price spike forecast with data mining techniques. *Electric Power Systems Research*, 73(1):19 – 29, 2005. ISSN 0378-7796. doi: http://dx.doi.org/10.1016/j.epsr.2004.06.002. URL `http://www.sciencedirect.com/science/article/pii/S0378779604001257`.

[16] Nima Amjady and Farshid Keynia. A new prediction strategy for price spike forecasting of day-ahead electricity markets. *Applied Soft Computing*, 11(6):4246 – 4256, 2011. ISSN 1568-4946. doi: http://dx.doi.org/10.1016/j.asoc.2011.03.024. URL `http://www.sciencedirect.com/science/article/pii/S1568494611001207`.

[17] Hsiao-Tien Pao. Forecasting electricity market pricing using artificial neural networks. *Energy Conversion and Management*, 48(3):907 – 912, 2007. ISSN 0196-8904. doi: http://dx.doi.org/10.1016/j.enconman.2006.08.016. URL `http://www.sciencedirect.com/science/article/pii/S0196890406002500`.

[18] Dawit Hailu Mazengia and LA Tuan. Forecasting spot electricity market prices using time series models. In *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on*, pages 1256–1261. IEEE, 2008.

[19] Jose Alvarez-Ramirez and Rafael Escarela-Perez. Time-dependent correlations in electricity markets. *Energy Economics*, 32(2):269 – 277, 2010. ISSN 0140-9883. doi: http://dx.doi.org/10.1016/j.eneco.2009.05.008. URL `http://www.sciencedirect.com/science/article/pii/S0140988309000814`.

[20] Ronald Huisman and Ronald Mahieu. Regime jumps in electricity prices. *Energy Economics*, 25 (5):425 – 434, 2003. ISSN 0140-9883. doi: http://dx.doi.org/10.1016/S0140-9883(03)00041-0. URL `http://www.sciencedirect.com/science/article/pii/S0140988303000410`.

[21] Ronald Huisman, Christian Huurman, and Ronald Mahieu. Hourly electricity prices in day-ahead markets. *Energy Economics*, 29(2):240 – 248, 2007. ISSN 0140-9883. doi: http://dx.doi.org/10.1016/j.eneco.2006.08.005. URL `http://www.sciencedirect.com/science/article/pii/S0140988306000995`.

[22] A. Boogert and D. Dupont. When supply meets demand: The case of hourly spot electricity prices. *Power Systems, IEEE Transactions on*, 23(2):389–398, May 2008. ISSN 0885-8950. doi: 10.1109/TPWRS.2008.920731.

[23] Robert JW de Groot, Bram MJ Vonk, Hans JA Beckers, and Johannes G Slootweg. Development of a charge path optimization controller block for a battery energy storage system. 2014.

[24] R.J.W. de Groot, P. Karaliolios, J.G. Slootweg, E.C.W. de Jong, and F. Van Overbeeke. Smart storage: Embedding local storage in the distribution grid. In *PowerTech (POWERTECH), 2013 IEEE Grenoble*, pages 1–6, June 2013. doi: 10.1109/PTC.2013.6652142.

[25] COMSOL. The joule heating effect, 2015. URL `http://www.comsol.com/multiphysics/the-joule-heating-effect`. [Online; accessed 16-August-2015].

[26] electrical4u. Joules law of heating, 2015. URL `http://www.electrical4u.com/joules-law/`. [Online; accessed 16-August-2015].

[27] R.A. Millikan, E.S. Bishop, and American Technical Society. *Elements of Electricity: A Practical Discussion of the Fundamental Laws and Phenomena of Electricity and Their Practical Applications in the Business and Industrial World*. American Technical Society, 1917. URL https://books.google.nl/books?id=dZM3AAAAMAAJ.

[28] Liander. Open data, 2013. URL https://www.liander.nl/over-liander/innovatie/open-data/data.

[29] Gemeente Schiedam. Inkoop elektriciteit 2015-2016 vp&msv gemeenten. TenderNed, May 2014. URL https://www.tenderned.nl/tenderned-web/aankondiging/detail/documenten/akid/959cb65d173f52ef99fe40933ca86201/pageId/D909B/huidigemenu/aankondigingen/da/false/cid/245764/cvp/join.

[30] NEDU. Verbruiksprofielen. NEDU, May 2015. URL http://nedu.nl/portfolio/verbruiksprofielen/.

[31] Solar is Future. What does kilowatt peak (kwp) actually mean?, 2015. URL http://www.solar-is-future.com/faq-glossary/faq/photovoltaic-technology-and-how-it-works/what-does-kilowatt-peak-kwp-actually-mean/. [Online; accessed 28-May-2015].

[32] Moraitis Panagiotis. *Review of the Operation Performance of Grid-Connected PV Systems*. PhD thesis, Utrecht University, 2014.

[33] Stan van Hoesel. *Models and Algorithms for Single-item Lot Sizing Problems*. PhD thesis, Erasmus Universiteit Rotterdam, 1991.

[34] Harvey M Wagner and Thomson M Whitin. Dynamic version of the economic lot size model. *Management science*, 5(1):89–96, 1958.

[35] Albert Wagelmans, Stan Van Hoesel, and Antoon Kolen. Economic lot sizing: an o (n log n) algorithm that runs in linear time in the wagner-whitin case. *Operations Research*, 40(1-supplement-1):S145–S156, 1992.

[36] Laurence A. Wolsey. Solving multi-item lot-sizing problems with an mip solver using classification and reformulation. *Management Science*, 48(12):1587–1602, 2002. doi: 10.1287/mnsc.48.12.1587.442. URL http://dx.doi.org/10.1287/mnsc.48.12.1587.442.

[37] Hsin-Der Chen, DonaldW. Hearn, and Chung-Yee Lee. A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *Journal of Global Optimization*, 4(3):285–300, 1994. ISSN 0925-5001. doi: 10.1007/BF01098363. URL http://dx.doi.org/10.1007/BF01098363.

[38] Behrooz Karimi, SMT Fatemi Ghomi, and JM Wilson. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5):365–378, 2003.

[39] Richard A Sandbothe and Gerald L Thompson. Decision horizons for the capacitated lot size model with inventory bounds and stockouts. *Computers & operations research*, 20(5):455–465, 1993.

[40] J. Gutiérrez, A. Sedeño Noda, M. Colebrook, and J. Sicilia. A new characterization for the dynamic lot size problem with bounded inventory. *Comput. Oper. Res.*, 30(3):383–395, March 2003. ISSN 0305-0548. doi: 10.1016/S0305-0548(01)00105-8. URL http://dx.doi.org/10.1016/S0305-0548(01)00105-8.

[41] Stephen F. Love. Bounded production and inventory models with piecewise concave costs. *Management Science*, 20(3):313–318, 1973. doi: 10.1287/mnsc.20.3.313. URL http://dx.doi.org/10.1287/mnsc.20.3.313.

[42] Xiao Liu, Feng Chu, Chengbin Chu, and Chengen Wang. Some new properties for capacitated lot sizing problem with bounded inventory and stockouts. *WSEAS Transactions on Mathematics*, 3:65–70, 2004.

[43] M. Fush. Marktubersicht batteriesysteme. *photovoltaik*, 10:46–49, October 2012.

[44] Andrei Shleifer and Robert W. Vishny. The limits of arbitrage. *The Journal of Finance*, 52(1):35–55, 1997. ISSN 1540-6261. doi: 10.1111/j.1540-6261.1997.tb03807.x. URL http://dx.doi.org/10.1111/j.1540-6261.1997.tb03807.x.

[45] Peter Bro Miltersen. The min cost flow problem. Aarhus University, February 2007.

[46] Andrew V Goldberg and Robert E Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM (JACM)*, 36(4):873–886, 1989.

[47] R.E. Tarjan. *Data Structures and Network Algorithms*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1983. ISBN 9781611970265. URL https://books.google.nl/books?id=dJCxyztJOEMC.

[48] J.M. van den Akker, S.L. Leemhuis, and G.A. Bloemhof. Optimizing storage placement in electricity distribution networks. In Stefan Helber, Michael Breitner, Daniel Rösch, Cornelia Schön, Johann-Matthias Graf von der Schulenburg, Philipp Sibbertsen, Marc Steinbach, Stefan Weber, and Anja Wolter, editors, *Operations Research Proceedings 2012*, Operations Research Proceedings, pages 183–188. Springer International Publishing, 2014. ISBN 978-3-319-00794-6. doi: 10.1007/978-3-319-00795-3_27. URL http://dx.doi.org/10.1007/978-3-319-00795-3_27.

[49] Clayton R Paul. *Fundamentals of Circuit Analysis*. Wiley, 2000.

[50] G.J. ter Haar. Waardebepaling kleine ondergrondse infrastructuur. Technical report, Grontmij, October 2995. URL http://www.grontmij.nl/Diensten/Bodem/Ondergrondse-infrastructuur/Documents/Rapport%20Waardebepaling%20kleine%20ondergrondse%20infrastructuur.pdf.

[51] T. Seba. *Clean Disruption of Energy and Transportation: How Silicon Valley Will Make Oil, Nuclear, Natural Gas, Coal, Electric Utilities and Conventional Cars Obsolete by 2030*. Jazzybee Verlag, 2014. ISBN 9780692210536. URL https://books.google.nl/books?id=xRdMBAAAQBAJ.

[52] RVO. Cijfers elektrisch vervoer, 2015. URL http://www.rvo.nl/onderwerpen/duurzaam-ondernemen/energie-en-milieu-innovaties/elektrisch-rijden/stand-van-zaken/cijfers. [Online; accessed 18-August-2015].

[53] Gabriel R. Bitran and Horacio H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):pp. 1174–1186, 1982. ISSN 00251909. URL http://www.jstor.org/stable/2630946.

[54] F.N. Claessen. *Smart grid control; An analysis of control systems within a Dutch residential microgrid incorporating decentralised renewable energy resources*. PhD thesis, Universiteit Utrecht, 2015.

[55] M. Florian, J. K. Lenstra, and A. H. G. Rinnooy Kan. Deterministic production planning: Algorithms and complexity. *Management Science*, 26(7):pp. 669–679, 1980. ISSN 00251909. URL http://www.jstor.org/stable/2630701.

[56] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman, 2002.

[57] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL http://www.gurobi.com.

[58] Stan van Hoesel, Albert Wagelmans, and Bram Moerman. Using geometric techniques to improve dynamic programming algorithms for the economic lot-sizing problem and extensions. 75(2):312–331, 1994. URL http://EconPapers.repec.org/RePEc:eee:ejores:v:75:y:1994:i:2:p:312-331.

[59] Netbeheer Nederland. Netbeheer voor u; storingen, 2015. URL http://www.netbeheernederland.nl/netbeheer-voor-u/storingen/. [Online; accessed 28-July-2015].

[60] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. *lpsolve : Open source (Mixed-Integer) Linear Programming system.*

Appendix A

Glossary

Amsterdam Power Exchange (APX) Wholesale electricity market operator in the Netherlands. Unless otherwise specified, this refers to the Day-Ahead market of the APX. 12

APX Amsterdam Power Exchange (APX). 12, 49–57, 59, 60, 78, 101

DSO Distributed System Operator. 1, 2, 6, 12, 20, 23

EDSN Energy Data Services Netherlands. 20, 21, 23

Energy Data Services Netherlands Company that analyses the use of electricity in the Netherlands. 20

node All entities that produce and/or consume electricity in electricity grids. Examples include households, offices, traditional power plants, solar installation or a combination of multiple entities. 13, 65, 77–79

PV Photo Voltaic. 1, 23

RES Renewable Energy Sources. 1

SoC State of Charge. 71

SSU Smart Storage Unit. 12

State of Charge Battery charge as percentage of the maximum battery capacity. 71

Storage-Cable-Node model Simple electricity grid consisting of a single storage or battery B, a node W and cable c connecting B and W with the external grid. 13–16, 50, 76

TenneT Transmission system operator in the Netherlands. 6–8, 12, 14, 45, 49–55, 57, 70, 78, 100–102, 104

TSO Transmission System Operator. 6–8, 23, 90

TTh Unit for cable damage using the temperature model for cable damage.. 66, 80

B List of Figures

Appendix C

Supplementary Figures

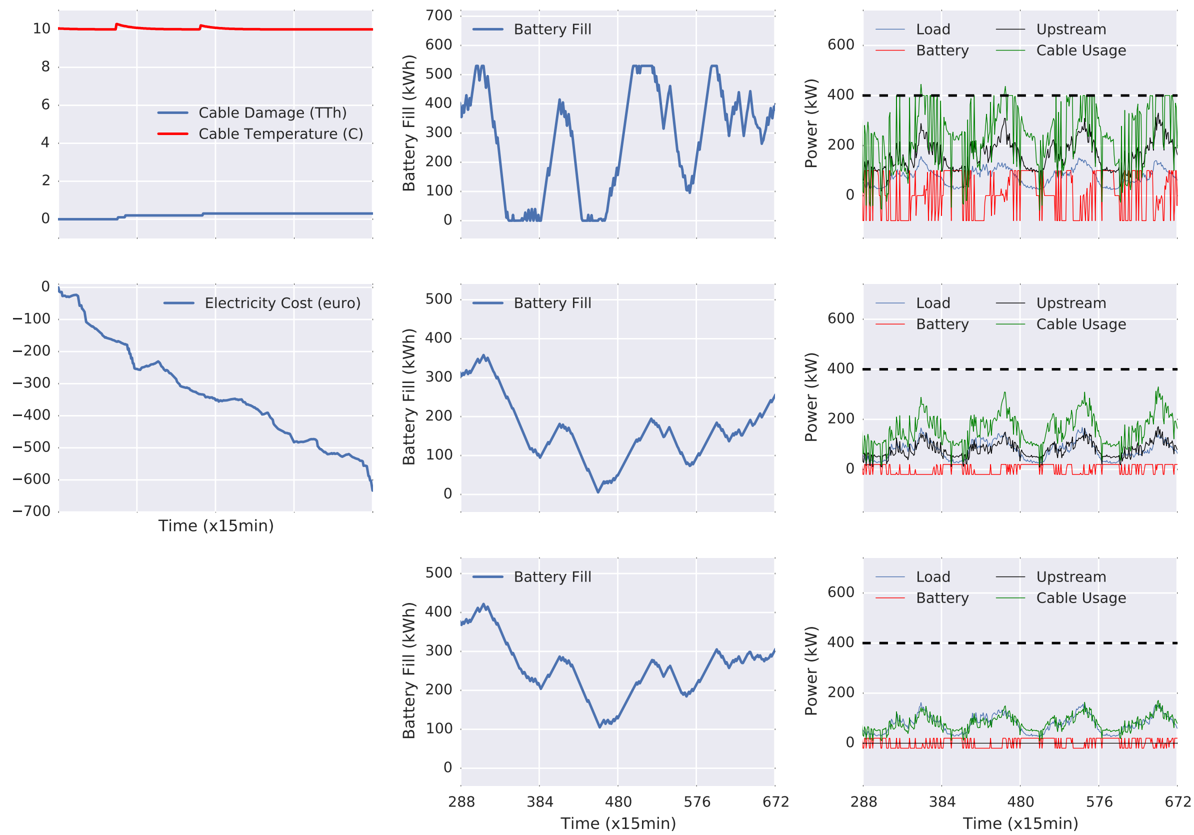


Figure C.1: The altered linear program definition in Section 8.2 allows for simulations of more complex grid structures. The rows depict the battery charge and charge rates of B_1, B_2 and B_3, respectively.

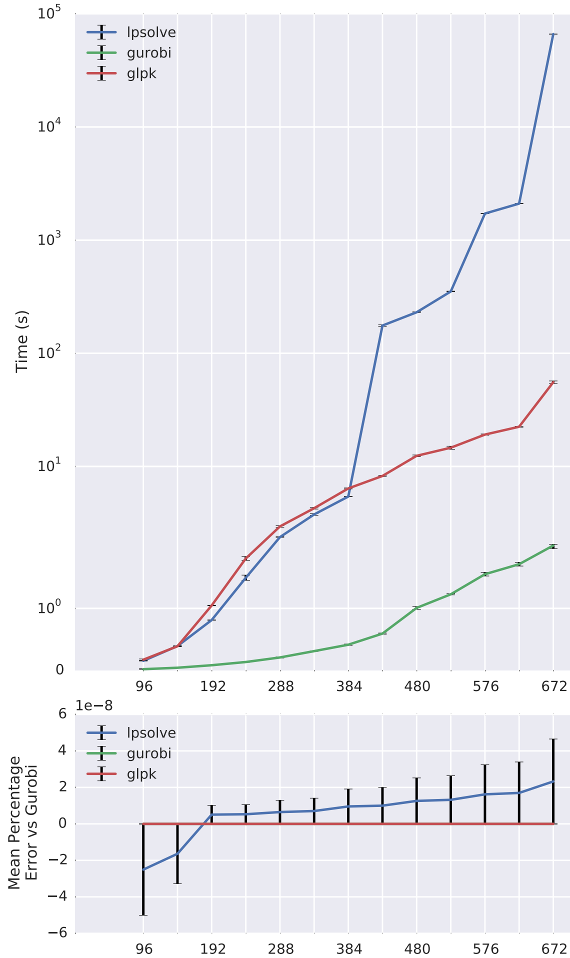


Figure C.2: Mixed Integer Programming (MIP) comparison of the Gurobi, LPsolve and GLPK optimisers. Top: computation time versus planning length (# intervals). Bottom: deviation in objective function from the Gurobi results. Note that the $T = 672$ calculation took too long to finish in the LPsolve tests.

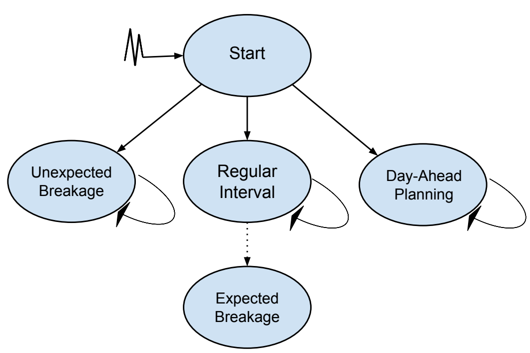


Figure C.3: Simulation Event Graph of the model that was developed as part of this study (See Section 6.2.4).

Appendix D

Supplementary Data

D.1 Data Overview

Name	# Scenarios	# Entities	Year	Date Range	Resolution	# Data Points
Load Demand						
Zonnedael	1	75	2013	1 year	15-minute	2,628,00
Schiedam	1	69	2013	1 year	15-minute	2,417,760
EDSN	1	9	2012-2015	4 years	15-minute	1,261,440
Claessens	5	6	?	1 year	15-minute	1,051,200
PV Production						
Solar Log	1	575	2011-2015	4 years	1-day	515169
Moraitis	1	1	2013-2014	10 months	5-minute	45,263
Market Prices						
APX Market Price	1	1	2007,2010-2014,2030	7 years	15-minute	525,600
TenneT Imbalance Price	1	2	2010-2014	5 years	15-minute	175,296

Table D.1: Overview of of all data used in this study.

D.2 Additional Data Sources

The following data sets are not featured in this study. The data was collected and visualised here perhaps to fulfil a role in another study.

D.2.1 Claessen [54] Simulated Domestic Load Data

The Claessen [54] Simulated Domestic Load Data contains a projection of future electricity use of households in the Netherlands. The data was simulated as part of Claessen [54]. See Figure D.1 for a characterisation of the data.

D.2.2 SKO Television Viewership Data

On request, the Stichting Kijkcijfer Onderzoek (SKO) has provided daily total viewer-ship data (separated in daily and evening totals) for every day in 2002 until 2015. This data can answer the question whether

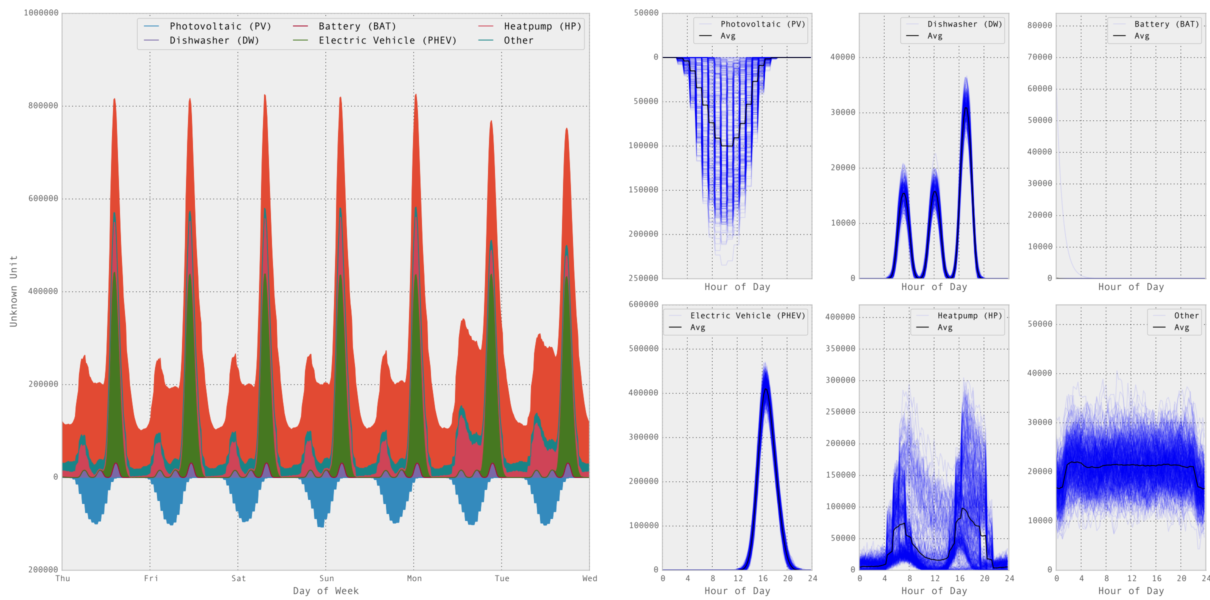


Figure D.1: Example Claessen [54] scenario '*No Control Optimistic*'. **(A)** Average Weekly Production/-Consumption Patterns (cumulative) of different parts of the simulated household. **(B)** Average Daily Load Patterns of different parts of the simulated household. Blue lines reflect individual weeks, the black line marks the average load.

massive television use could be used to build better load forecasting models. For instance, in the Netherlands the broadcasts of games played by the national soccer league are very popular. In both 2010 and 2014 the top 3 television events were games played by the Netherlands at the World Cup. This massive interest in television could lead to a particular energy demand. We therefore propose the research question whether television viewer-ship predictions allow for more accurate residential load forecasting models.

D.2.3 Grid Frequency

A data set was obtained containing 4-second interval data from a 3-month period in 2014 with grid frequencies in the Dutch electricity grid. In most countries, a 50 Hz alternating current is used. Deviations from these 50 Hz arise from a surplus or shortage in electricity production, and can cause damage or even blackouts. Therefore, the TSO is responsible for monitoring this frequency, or even producing or consuming electricity in order to balance the system. For more information on the TSO and grid imbalance, refer to Section 2.2.

In Figure D.3 a sampling of the data is shown as a carpet plot. The frequency heat-map shows a number of interesting phenomena. For instance, on October 26th 02:00 there was a change from summer to wintertime. This created a surplus of energy, resulting in a relatively high frequency. This may be due to system errors, where producing or consuming units were scheduled at the wrong hour. Furthermore, a clear period of relatively high imbalance is shown at the start and end of an ordinary work day (namely at 6:00-8:00 as well as 17:00-19:00). Although we can only guess to what causes this imbalance, we speculate this might have to do with the unpredictability in precise office and home hours (caused by for instance traffic delay). Lastly, we can distinguish changes in the frequency at the start/end of hours. This is likely due to the fact that day-ahead APX market (see Section 2.2) go by hourly periods. As a result, at the start and end of each hour machines will be turned on/off. This leads to the imbalance, as these on/off switches do not occur in sync.

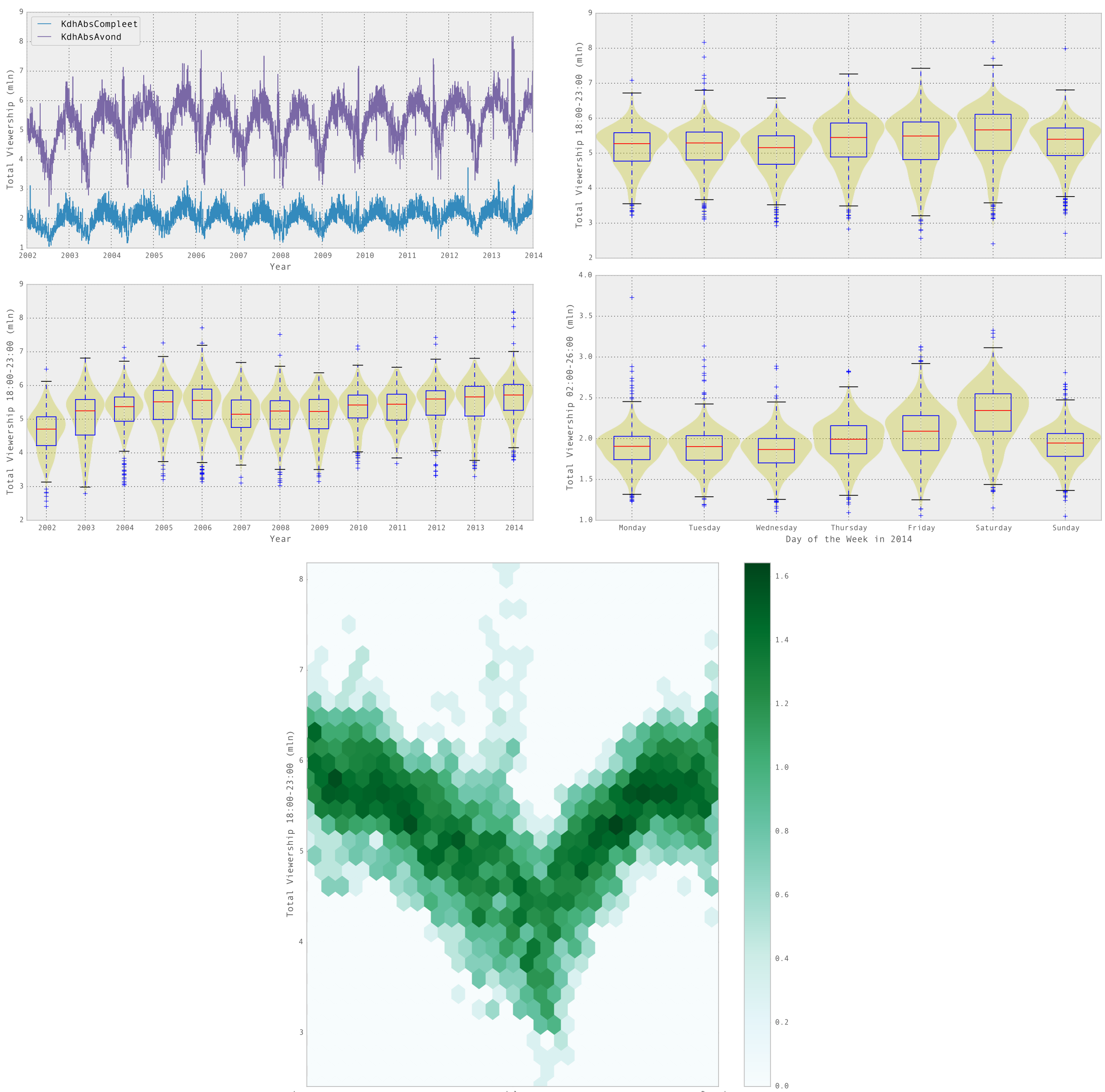


Figure D.2: **(A)** (top) Television Viewership per Year during the Day (*KdhAbsCompleet*) and in the Evening (*KdhAbsAvond*). (bottom) Violin Plot of Evening Viewership per Year. plot**(B)** Television Viewership per Day of the Week in the Evening (top) and during the Day (bottom). **(C)** Hexagonal Bining Plot of Viewership in 2002-2015 per day of the year.

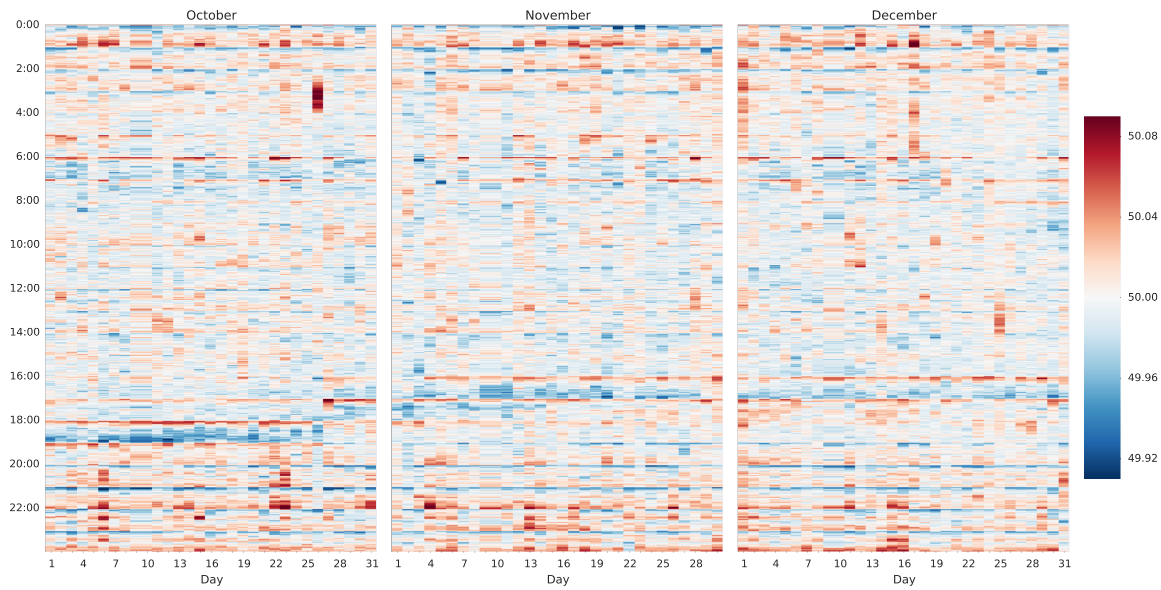


Figure D.3: Sampled Grid Frequency (Hz) over a 3 month period in the Dutch electricity network. A value for every 100 seconds is shown.

Appendix E

Code and Scripts

E.1 Java implementation of Ramezani et al. [3] H_2 heuristic

```java
import java.util.Arrays;
import util.ArrayIndexComparator;


public class RamezaniHeuristic {

	public static double[] run(int duration, double[] priceForecasts,	double[] loadForecasts,
			double batteryChargeSpeedMinimum, double batteryChargeSpeedMaximum, double batteryCapacityMaximum,
			double batteryFillZero, double cableCapacityMaximum, double batteryEfficiency) {

		// 0. Preparations
		int n = duration;

		double Cm = cableCapacityMaximum;
		double Bm = batteryCapacityMaximum;
		double BCmax = batteryChargeSpeedMaximum;
		double BCmin = batteryChargeSpeedMinimum;
		double bEff = batteryEfficiency;
		double bZero = batteryFillZero;
		double[] prices = priceForecasts;

		double averagePrice = 0;
		for (double price : priceForecasts) averagePrice += price;
		averagePrice /= n;

		boolean[] isCritical = new boolean[n];
		for (int i=0; i<n; i++) isCritical[i] = Math.abs(loadForecasts[i]) > Cm;

		ArrayIndexComparator priceComparator = new ArrayIndexComparator(priceForecasts);
		Integer[] priceSortedIndexes = priceComparator.createIndexArray();
		Arrays.sort(priceSortedIndexes, priceComparator);

		// 1. Make Wishlist

		double[] wishlist = new double[n];

		for (int i=0; i<n; i++) {

			if (Cm < loadForecasts[i]) {
				wishlist[i] = Math.max(BCmin, Cm - loadForecasts[i]);
			} else if (-Cm > -Math.abs(loadForecasts[i])) {
				wishlist[i] = Math.min(BCmax, - loadForecasts[i] - Cm);
			} else {

				if (priceForecasts[i] < averagePrice) { // We want to buy (charge)

					wishlist[i] = Math.min( BCmax, Cm - loadForecasts[i]);

				} else { // We want to sell

					wishlist[i] = Math.max( BCmin, - Cm - loadForecasts[i]);

				}
			}

		}

		int i = 0;
		int j;

		double currentCharge, blockCharge;

		while (i < n) {

			currentCharge = (i>0) ? charges(wishlist, bZero, bEff)[i-1] : bZero;

			if (!isCritical[i]) {


				// i is non-critical
```

```java
            // find same-sign non-critical block

            j = i;
            blockCharge = 0;
            while (true) {
                // Iterate j until critical block or different sign
                blockCharge += wishlist[j] * (wishlist[j]>0 ? batteryEfficiency : 1.);
                j++;
                if (j >= n || ((Math.signum(wishlist[i])!=Math.signum(wishlist[j])) || isCritical[j])) {
                    break;
                }
            }

            if (blockCharge+currentCharge < 0 || blockCharge+currentCharge > Bm) {

                double[] block = new double[j-i];
                for (int k=0; k<(j-i); k++) block[k] = priceForecasts[i+k];

                ArrayIndexComparator comparator = new ArrayIndexComparator(block);
                Integer[] indexes = comparator.createIndexArray();
                Arrays.sort(indexes, comparator);

                if (blockCharge+currentCharge < 0) {

                    for (int k=0; k<block.length; k++) {

                        if (currentCharge + blockCharge - wishlist[indexes[k]+i] < 0) {
                            blockCharge -= wishlist[indexes[k]+i];
                            wishlist[indexes[k]+i] = 0;
                        } else {
                            wishlist[indexes[k]+i] -= (blockCharge+currentCharge);
                            break;
                        }

                    }

                    currentCharge = 0;

                } else if (blockCharge+currentCharge > Bm) {

                    double toReduce = (blockCharge + currentCharge) - Bm;
                    for (int k=block.length-1; k>=0; k--) {

                        if (toReduce - (wishlist[indexes[k]+i]*bEff) > 0) {
                            toReduce -= wishlist[indexes[k]+i] * bEff;
                            wishlist[indexes[k]+i] = 0;
                        } else {
                            wishlist[indexes[k]+i] -= (1/bEff)*toReduce;
                            break;
                        }

                    }

                    currentCharge = Bm;

                }

            } else { // Keep the block as is!
                currentCharge += blockCharge;
            }

            if (j==n) break; // Reached the end of the schedule

            i = j; // Move to the next block

            if (!isCritical[i]) continue; // For a non-critical block, move to step 2a
        }

        // Find same-sign critical block length and charge
        j = i;
        blockCharge = 0;
        while (true) {
            // Iterate j until critical block or different sign
            blockCharge += wishlist[j] * (wishlist[j]>0 ? bEff : 1.);
            j++;
            if (j >= n || ((Math.signum(wishlist[i])!=Math.signum(wishlist[j])) || !isCritical[j])) {
                break;
            }
        }

        if ((blockCharge+currentCharge < 0 || blockCharge+currentCharge > Bm) && i>1) {

            double[] charges = new double[i];
            charges[0] = batteryFillZero + wishlist[0] * (wishlist[0]>0 ? batteryEfficiency : 1);
            for (int k=1; k<i; k++) {
                charges[k] = charges[k-1] + wishlist[k] * (wishlist[k]>0 ? batteryEfficiency : 1);
            }

            if (blockCharge+currentCharge < 0) {

                double toAdd = Math.min(- (currentCharge+blockCharge), Bm - currentCharge);

                for (int k=0; k<n; k++) {

                    if (priceSortedIndexes[k] >= i) continue;

                    int k_ = priceSortedIndexes[k];

                    if (wishlist[k_] >= 0) continue;
                    if (isCritical[k_]) continue;

                    // Remove as possible from this step, consider:
                    // - charge up to then (can remove up what's
```

```java
                        // - cable

                        double newWish = Math.min(0, wishlist[k_] + toAdd);

                        // Check feasibility of this new wish (in the entire interval before i)
                        double[] chargesCopy = new double[i];
                        System.arraycopy(charges, 0, chargesCopy, 0, i);
                        boolean feasible = true;
                        chargesCopy[k_] = charges[k_] + (newWish - wishlist[k_]);
                        for (int q=k_+1; q<i; q++) {
                            chargesCopy[q] = chargesCopy[q-1] + wishlist[q] * (wishlist[q]>0 ? bEff : 1);
                            if (chargesCopy[q] < 0 || chargesCopy[q] > Bm) {
                                feasible = false;
                                break;
                            }
                        }

                        if (feasible) {
                            System.arraycopy(chargesCopy, k_, charges, k_, i-k_);
                            toAdd -= newWish - wishlist[k_];
                            wishlist[k_] = newWish;

                            currentCharge = chargesCopy[i-1];
                        }

                        if (toAdd <= 0) break;
                    }


                } else if (blockCharge+currentCharge > Bm) {

                    double toReduce = Math.min(currentCharge - (Bm-blockCharge), currentCharge);

                    for (int k=n-1; k>=0; k--) {

                        if (priceSortedIndexes[k] >= i) continue;

                        int k_ = priceSortedIndexes[k];

                        if (wishlist[k_] <= 0) continue;
                        if (isCritical[k_]) continue;

                        // Remove as possible from this step, consider:
                        // - charge up to then (can remove up what's
                        // - cable

                        double smallestNewWish = Math.max(0, (1/bEff) * ((wishlist[k_]*bEff) - toReduce));

                        // Check feasibility of this new wish (in the entire interval before i)
                        double[] chargesCopy = new double[i];
                        System.arraycopy(charges, 0, chargesCopy, 0, i);
                        boolean feasible = true;
                        chargesCopy[k_] = charges[k_] + (smallestNewWish - wishlist[k_])*bEff;
                        for (int q=k_+1; q<i; q++) {
                            chargesCopy[q] = chargesCopy[q-1] + wishlist[q] * (wishlist[q]>0 ? batteryEfficiency : 1);
                            if (chargesCopy[q] < 0 || chargesCopy[q] > Bm) {
                                feasible = false;
                                break;
                            }
                        }

                        if (feasible) {
                            System.arraycopy(chargesCopy, k_, charges, k_, i-k_);
                            toReduce += (smallestNewWish - wishlist[k_])*bEff;
                            wishlist[k_] = smallestNewWish;

                            currentCharge = chargesCopy[i-1];
                        }
                        if (toReduce <= 0) break;

                    }

                }

                // new currentCharge
                if (blockCharge+currentCharge >= 0 && blockCharge+currentCharge < Bm) {
                    // Allow the wishes in this critical block (leave unchanged)
                    currentCharge += blockCharge;
                } else if (blockCharge+currentCharge < 0)  {
                    // Divide the current charge evenly over this block
                    for (int k=i; k<j; k++) {
                        wishlist[k] = - (currentCharge / (j-i));
                    }
                    currentCharge = 0;
                } else if (blockCharge+currentCharge > Bm) {
                    // Divide the remaining charge space evenly over this block
                    for (int k=i; k<j; k++) {
                        wishlist[k] = ((Bm-currentCharge)*(1/bEff)) / (j-i);
                    }
                    currentCharge = Bm;
                }

            } else { // Keep the block as is!
                currentCharge += blockCharge;
            }

            i = j;

        }

        return wishlist;

    }
```

```
	private static double[] charges(double[] arr, double zero, double eff) {
		double[] charges = new double[arr.length];
		charges[0] = zero + arr[0] * (arr[0]>0 ? eff : 1);
		for (int k=1; k<arr.length; k++) {
			charges[k] = charges[k-1] + arr[k] * (arr[k]>0 ? eff : 1);
		}
		return charges;
	}

}
```

E.2 Python script for Solar-Log data collection

```
import re, sys, os, time, datetime, urllib2
from datetime        import timedelta, date
from bs4        import BeautifulSoup

DAILY_YIELD_RANGE=(5*365)+220

address="http://home.solarlog-web.nl/sds/modul/SolarLogReference2/_sysreturn.php?m=SolarLogReference2"
adress+="&=s=8a0712a866935855021676cfa2b494ec&cid=5&action=result&mo=result&cc=&pc=&mk=&of=&ot=&rf=&rt=&pf=&pt=&cy=&wp=0&wd=0&ws=0&cp="

##

Number_Pages=urllib2.urlopen(str(address)).read()
First_Soup=BeautifulSoup(Number_Pages)
Number_1=First_Soup.find('div',{'class':'footer'})
Number_List=re.findall(r'\d+', Number_1.text)

##
for i in range(int(sys.argv[1]),int(sys.argv[1])+10):
	print "PAGE: ", i

	webpage=urllib2.urlopen(address+str(i)).read()
	soup=BeautifulSoup(webpage)

	for Installations in soup.findAll('div', {'style':"display:inline;float:left;"}):
		x=Installations.find('td',{'style':"width:316px;background:url('/sds/modul/SolarLogReference2/images/s_mitte_blauer_balken.png') repeat-x;height:19px;overflow:hidden;padding:1px 0px 0px 4px;color:#FFF;font-size:12px;"})
		if x == None:
			continue

		for Names in Installations.find('td',{'style':"width:316px;background:url('/sds/modul/SolarLogReference2/images/s_mitte_blauer_balken.png') repeat-x;height:19px;overflow:hidden;padding:1px 0px 0px 4px;color:#FFF;font-size:12px;"}):
			Installation_Name=Names.text
			print "name="+Installation_Name
			if Installation_Name == "***":
				print "skipping anonymous installation"
				continue
			if Installation_Name == "":
				print "installation name is required"
				continue
			filename = "installation_" + Installation_Name
			for char in [' ', ',', '.', '/']: filename = filename.replace(char, '')
			filename += ".txt"

		if os.path.isfile(filename):
			print "Skipping the installation with file " + filename+ ", file exists"
			continue

		Installation_File = open(filename, 'w')
		Installation_File.write('Name='+Installation_Name.encode('utf-8')+'\n')

		# Baujahr - Installation Date
		# Standort - Location
		# Anlagenleistung - System Size
		# Wechselrichter - Inverter
		# Module - Installation Panel
		# Ausrichtung - Orientation
		# Dachneigung - Slope
		#for Locations in Installations.findAll('div', attrs={'title':re.compile('')}): print Locations

		for entry in Installations.findAll('div', {'title':re.compile('Baujahr:')}):
			entry_value=entry.text
			Installation_File.write('Date='+entry_value.encode('utf-8')+'\n')
		for Location in Installations.findAll('div', attrs={'title':re.compile('Standort:')}):
			Installation_Location=Location.text
			Installation_File.write('Location='+Installation_Location.encode('utf-8')+'\n')
		for Size in Installations.findAll('div', {'title':re.compile('Anlagenleistung:')}):
			Installation_Size=Size.text
			Installation_File.write('System Size='+Installation_Size.encode('utf-8')+'\n')
		for entry in Installations.findAll('div', {'title':re.compile('Wechselrichter:')}):
			entry_value=entry.text
			Installation_File.write('Inverter='+entry_value.encode('utf-8')+'\n')
		for entry in Installations.findAll('div', {'title':re.compile('Module:')}):
			entry_value=entry.text
			Installation_File.write('Panel='+entry_value.encode('utf-8')+'\n')
		for entry in Installations.findAll('div', {'title':re.compile('Ausrichtung:')}):
			entry_value=entry.text
			Installation_File.write('Orientation='+entry_value[:-2].encode('utf-8')+'\n')
		for entry in Installations.findAll('div', {'title':re.compile('Dachneigung:')}):
			entry_value=entry.text
			Installation_File.write('Slope='+entry_value[:-2].encode('utf-8')+'\n')


		# Additional Details

		for link in Installations.findAll('a',{'style':"text-decoration:none;color:black;cursor:pointer;line-height:20px;vertical-align:top;"}):
```

```
            pvlink=(link['href'])

            print "Obtaining data from page ", pvlink

            Data_PV_link=re.findall(r'\d+', pvlink)
            Full_Link="http://home.solarlog-web.nl/"+str(int(Data_PV_link[-1])-1)+".hrml"
            Foul_Link_URL=urllib2.urlopen(Full_Link)
            Soup_DataLink=BeautifulSoup(Foul_Link_URL)
            for Two_Headings in Soup_DataLink.findAll('td', {'width':'663'},{'class':'headline'}):
                Module_Type=Two_Headings.next_element
                for i in range(19):
                    Module_Type=Module_Type.next_element
                if Module_Type is not None:
                    try:
                        Installation_File.write(Two_Headings.text+"Count="+Module_Type.text+"\n")
                        Installation_File.write(Two_Headings.text+"Type="+Module_Type.next_sibling.text+"\n")
                    except AttributeError:
                        print "No Available Data"


            # Yield Data
            pv_linkslocation_2=urllib2.urlopen(pvlink)
            soup=BeautifulSoup(pv_linkslocation_2)
            statistik=soup.findAll('script')
            #print re.search(r'(var svgsrc = ")(http|ftp|https):\/\/([\w\-_]+(?:(?:\.[\w\-_]+)+))([\w\-\.,@?^=%&amp;:/~\+#]*[\w\-\@?^=%&amp;/~\+#])?', str(statistik)).group()

            next_adress=re.search(r'(var svgsrc = ")(http|ftp|https):\/\/([\w\-_]+(?:(?:\.[\w\-_]+)+))([\w\-\.,@?^=%&amp;:/~\+#]*[\w\-\@?^=%&amp;/~\+#])?', str(statistik))
            help_me=next_adress.group()
            last_step=re.search(r'(http|ftp|https):\/\/([\w\-_]+(?:(?:\.[\w\-_]+)+))([\w\-\.,@?^=%&amp;:/~\+#]*[\w\-\@?^=%&amp;/~\+#])?', str(help_me))
            move_on=last_step.group()+"&flag=33"
            go_back=re.search(r'(http|ftp|https):\/\/([\w\-_]+(?:(?:\.[\w\-_]+)+))([\w\-\.,@?^=%&amp;:/~\+#]*[\w\-\@?^=%&amp;/~\+#])?offset=', str(move_on))
            go_back2=go_back.group()
            go_back3=re.search(r'&r=(\w*)&flag=(\w*)', str(move_on))
            go_back4=go_back3.group()

            print "going through %d dates for data" % DAILY_YIELD_RANGE

            for i in range(DAILY_YIELD_RANGE,0,-1):
                if (i%100==0): print "%d days remaining..." % i
                all_the_links=go_back2+'-'+str(i)+go_back4
                one_more_step=urllib2.urlopen(all_the_links)
                one_more_step2=BeautifulSoup(one_more_step)

                for find_ac in one_more_step2.findAll('g',{'transform':'translate(0 5)'}):
                    for guess_what in find_ac.findAll('text',{'text-anchor':'start'} ):
                        ac_value=guess_what.next_sibling
                        if ac_value is not None:
                            try:
                                dc_value=ac_value.next_sibling
                                #print (str(date.today()+timedelta(days=-i))+' '+guess_what.text+" " +ac_value.text+' '+dc_value.text+'\n')
                                Installation_File.write(str(date.today()+timedelta(days=-i))+' '+guess_what.text+"=" +ac_value.text+','+dc_value.text+'\n')
                            except AttributeError:
                                print "error"

            Installation_File.write("||end||\n")
            Installation_File.close()
```