Sustainable & smart distribution networks

Thesis

EE BEP Group B

Subgroup B.3 - Optimization Control

Tal Baranes
Rick Steman

TU Delft

Sustainable & smart distribution networks

Thesis

by

EE BEP Group B

to obtain the degree of Bachelor of Science
at the Delft University of Technology,

Students:	Tal Baranes (4968115)	
	Rick Steman (4953371)	
Project duration:	April 19, 2022 – June 24, 2022	
Thesis committee:	Dr. P.P. (Pedro) Vergara Barrios,	TU Delft, supervisor
	N. K. (Nanda) Panda,	TU Delft daily supervisor

TU Delft

Preface

In this thesis the build plans for a new residential neighbourhoord, Vechtrijk, are used as a basis for an hypothetical neighbourhood of the future. All data is converted to fit in this hypothetical neighbourhood with an abundance of PV systems, EV charging stations and electric heating inside the houses. While this hypothetical neighbourhood often simply is called 'the Vechtrijk neighbourhood', in fact it is not connected to the real build plans and always the hypothetical neighbourhood is meant. We thank the project developer, L. Roodbol from Blauwhoed , for sharing detailed maps from the build plans.

We thank our supervisor, Dr. P. P. Vergara and our daily supervisor, N. Panda for their continuing support.

EE BEP Group B Delft, June 2022

Contents

Nomenclature

$BESS$	Battery Energy Storage System
BMS	Battery Management System
COP	Coefficient of Performance
DG	Distributed Generation
DoD	Depth of Discharge
DSM	Demand Side Management
DSO	Distribution System Operator
EV	Electric vehicle
ILP	Integer Linear Program
LP	Linear Program
LV	Low Voltage
$MILP$	Mixed Integer Linear Program
MPC	Model Predictive Control
PEV	Plug-in Electric Vehicle
PV	Photovoltaic
SOC	State of Charge
SOE	State of Energy
$V2G$	Vehicle to Grid
$V2V$	Vehicle to Vehicle

1 Introduction

In the neighbourhood of the future, the residential and road transportation sectors will probably have shifted for a large part to all-electric solutions and many houses will have photovoltaic (PV) systems on their roof. This creates new challenges for grid operators because they introduce large fluctuations in the demand and production of electrical energy outside the control of the grid operator. Consequently, this could lead to congestion and problematic fluctuations in voltage and frequency in the distribution network. Grid congestion occurs when a grid overload prevents electricity from reaching the consumer. The project of this thesis is about designing a smart and sustainable grid without problems like congestion for an hypothetical neighbourhood of the future in the Netherlands. This hypothetical neighbourhood, which is based on plans for building an actual neighbourhood in Weesp, has an electric vehicle (EV) charging station at every parking spot and each house has PV systems and electric heating.

1.1. Analyses

Tackling global warming is seen by many as one of the biggest challenges humankind faces in the 21st century. Humans cause too much greenhouse gas emission, causing sunlight to warm up the earth's surface more through the greenhouse effect. This effect results in (possibly irreversible) changes in the climate worldwide, which could lead to catastrophic disasters. In the Netherlands, the residential sector produced 15.4 Tg CO_2 equivalent (CO_2 and other greenhouse gasses) in 2019 with heating, water heating, and cooking, which accounts for 8.5% of the total CO_2 equivalent emissions of the Netherlands in 2019. Fossil-fueled road transportation produced 29.6 Tg CO_2 equivalent, which accounts for 16.4% of the total CO_2 equivalent emission of the Netherlands in 2019 [1]. There are options like using electric cars, installing PV systems and electric heating and cooking to reduce the greenhouse gas emission in these sectors. However, these newly developed sustainable solutions have a drawback. The current low-voltage (LV) grid is not designed for their combined operation and the increasing penetration of PV systems, EV charging stations and electric heating and cooking could result in more problems like congestion and excessive fluctuations in voltage and frequency. In Amsterdam, congestion already is a significant problem [2].

Allowing more energy-sustainable options to be implemented and connected to the grid while preventing problems like congestion will be a challenge for engineers in the upcoming years. Upgrades to the grid and other solutions to cope with this challenge are researched and proposed in this thesis. Following this general introduction, first follows a section that elaborates on the neighbourhood of the future in general. Then a neighbourhood that is being developed right now in the Netherlands is introduced, this neighbourhood will be used as example neighbourhood of the future throughout the thesis. Finally congestion is explained, as coping with congestion is one of the biggest challenges that the neighbourhood of the future will create.

1.2. Neighbourhood of the future

In the neighbourhood of the future, the previously stated residential and road transportation sectors will probably have shifted for a large part to all-electric solutions. The advantage of all-electric solutions is that the source of energy is interchangeable, meaning that fossil fuels can be replaced by sustainable energy sources with a dramatically lower emission of greenhouse gasses like wind or solar energy. In the Netherlands electrical cooking and heating is quickly rising in popularity; the amount of heat extracted from the air with heat pumps for the heating of residential buildings has more than doubled between 2018 and 2020 [3]. Also, the number of EVs in the Netherlands has approximately doubled every year from 2017 until 2021 [4]. Finally, the number of PV systems installed on residential buildings have also been steadily increasing. It is easily visible when walking outside anywhere in the Netherlands that many households have PV systems on their roofs. While these solutions are effective in reducing our dependence on fossil fuels, they create new challenges for grid operators. First of all, because the grid is not designed for the high currents that will run through it, congestion will occur. Secondly, as part of the power generation is happening outside the control of the grid operators, it is harder to prevent problematic voltage and frequency fluctuations.

1.2.1. Vechtrijk neighbourhood

In order to design, forecast, and optimise a sustainable distribution network, first a neighbourhood needs to be considered. A new developing neighbourhood was found in Weesp in the Netherlands. It consists of 37 total houses, a combination of detached houses, semi-detached houses, terraced houses, and apartments, which can be seen in Figure 1.1. It also consists of a central parking area where electric vehicles might be parked and charged. We think this is a good representation of an all-encompassing neighbourhood in the Netherlands and thus a good neighbourhood to design a sustainable grid for [5].

Figure 1.1: The Vechtrijk neighbourhood ground plan

1.3. Congestion: Definition and issues

Grid congestion occurs when an overloaded grid prevents electricity from reaching the consumers. It can be illustrated as a kind of traffic jam, where the electrons in the wire are symbolized by the vehicles on a highway. During rush hour, too many vehicles are present and the flow of traffic may come to a halt. A similar situation occurs on power lines as demand or supply of power is in excess. The high current the power line has to carry can exceed the maximum capacity of the lines. This will eventually lead to power outages and costly repairs on power lines. In the Netherlands congestion is becoming a large issue. Figure 1.2 illustrates the significance of the problem. It is visible, that congestion due to high demand happens the most in urban areas such as the province of North Holland and around Leeuwarden. Furthermore, it is visible that congestion due to high generation happens more in rural areas such as the provinces Drenthe, Overijssel and Gelderland in the east of the Netherlands. In order to avoid power outages, Distribution System Operators (DSOs) disconnect parts of the system during power excess. This necessary measure has great impact; generation systems are put on standby or consumers will have to seize their activities. Additionally, the connection to the grid for a newly installed system might take more than a year [6].

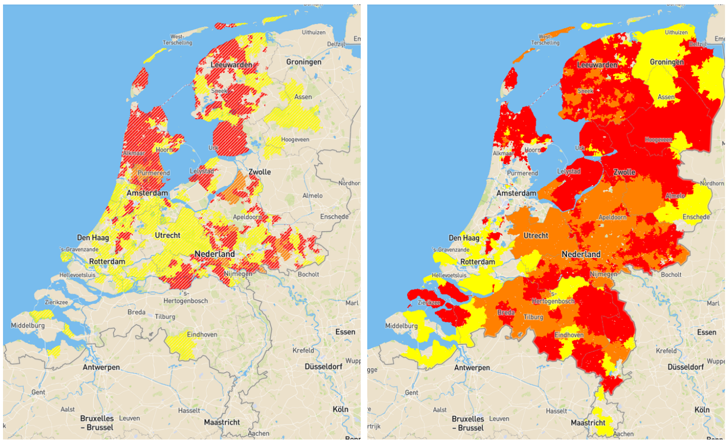


Figure 1.2: Consumption (left) and generation (right) congestion in The Netherlands [7]

1.4. Optimisation

When incorporating distributed generation (DG) in a neighbourhood, the chance of congestion increases rapidly. Therefore it is important to optimise the various sources, generations and storage to control the power flow within the LV grid of the neighbourhood. One of the essential resources when talking about DG is a battery energy storage system (BESS). This system manages and stores the excess energy generated by the DG and plays a crucial role to counteract the intermittent nature of DG. Nevertheless, BESS can be used for many more applications, like in the study of Datta [8], where BESS is designed to supply the extra demand of an EV charging station above the rated power of the transformer. Therefore, reducing the overloading on the transformer and reducing congestion. However, for any BESS to work, an accurate model must be created. Four types of energy storage models are mentioned in [9]: dynamic models, energy flow models, physics-based models, and black-box models. Within these models, the key parameters are power rating, energy capacity, efficiency and ramp rate. According to [10], a physics based model is necessary for accurately estimating the state of charge (SOC) for a battery management system (BMS), because the capacity of a battery changes as it decays. For lithium-ion batteries, models have been developed extensively, but most are too complicated for the calculations of a real-time BMS. A simplified model for a grid-level BMS is suggested by [10]. They also mention that improperly managed stacks of batteries can cause cell mismatch (a situation where not all battery cells are equally charged), which in turn can cause overcharging. This can be avoided by managing the batteries with a tiered architecture. The need for prediction for controlling BESSs and PVs is explained by [11]. They use model predictive control (MPC) to smooth out the power available from PVs and to match the user load and PV load better. The writers explain that the system becomes more adaptable and better at self-correcting when MPC is used. The prediction of the PV generation is built with an asymmetric least-squares fit on the PV output data and a linear error correction for uncertainties.

Another interesting approach for using batteries is to incorporate the EV battery into the system. Using personal vehicles as an energy storage solution for the grid has been discussed as far back as 1996 by Kempton [12]. A partial differential equation-based model to control the charging and discharging of a fleet of plug-in electric vehicles (PEV) is proposed by [13]. This model concentrates on optimising energy delivery to the grid (that is contracted ahead of time), supplying car users with sufficiently charged cars, and minimizing the costs of charging the vehicle batteries after use. Using the vehicle battery for the grid is called vehicle to grid (V2G) transfer. Another way to incorporate the vehicle batteries to reduce congestion is vehicle to vehicle (V2V) transfer [14]. By charging PEVs that need to be charged during peak times with vehicles that are still charged (for example, because the owners did not need their car that day), the amount of extra energy necessary from the external electric grid can be reduced. A direct DC vehicle to vehicle transfer is proposed in [15], to increase the efficiency of power transfer between electric vehicles.

For optimising an energy storage system, the authors in [9] argue that in power systems, mixed-integer linear programming (MILP) is often used as an extension of linear programming (LP). Since the extra decision variables in MILP are useful for applications with on and off states. The optimisation happens in two stages, one for the day-ahead market and another one for the real-time market, which is used to adapt to unpredicted changes. Other examples for optimisation are Robust optimisation for including worst-case scenario's into the optimisation, stochastic optimisation for including randomness and dynamic optimisation. However, dynamic optimisation requires vast amounts of coding and is slow as a result, and thus is not used for large scale systems.

Many studies also look at the possibility of changing the power consumption of the customer in such a way that congestion is eliminated. This so called demand side management (DSM) is used to decrease consumption, especially during peak hours [16]. This paper uses a profile steering methodology to steer the energy demand in a desired profile, which lowers the peaks. As a result this lowers the strain on the transformer. Another example of DSM can be found in [17]. In this paper, an optimisation algorithm is proposed to control domestic electricity and heat demand, as well as generation and storage. This has been modelled as an Integer Linear Program (ILP), with its objective being to minimize the costs while heat and electricity demand is matched using BESS, PV, and DSM. Another study takes it one step further and introduces demand prediction uncertainty [18]. Here, DSM under demand prediction uncertainty is used to reduce peaks. It uses non-linear programming to optimise the loads between buildings while also sharing energy between these buildings. In [19], the energy sharing and non-linear optimisation model used in [18] are created. This study analyzes the energy problem from a top-down approach (e.g., energy and battery sharing).
A similar approach is used in [20], although without using DSM. Here, houses with similar demand profiles are combined to increase sharing capabilities between clusters. These clusters were then modeled in an LP model to optimise their electricity usage. A different method is used in [21]. Namely particle swarm optimisation. They use this method to distribute the energy within the system, where the BESS provides the extra energy whenever the solar energy is not available and thus manages the congestion. Finally, [22] provides a mixed-integer non-linear programming model for optimising the operation of multiple buildings within a microgrid. By using linearization techniques, this model is then converted to a MILP model. The use of a MILP model for a microgrid can also be seen in [23]. This paper presents a multi-objective framework to quantify flexibility for a large business park in Eindhoven. The MILP model created in this paper is developed in Python Pyomo. LP models are often used since they can be solved by commercially available solvers and are faster at solving the model at the cost of accuracy. However, when dealing with high amounts of energy, slight accuracy imbalances are negligible.

1.5. Thesis objective

The objective of this thesis is to create a smart and sustainable distribution network. This will be achieved by subdividing the problem into three parts (which will be explained in Section 1.6). The objective of this specific thesis is to create an optimisation model to control the powerflow in the distribution network. Most studies focus mainly on the economic aspect of optimisation and thus consider the price of electricity as their objective function to minimize. This thesis, however focuses on reducing the grid interaction and as a result, minimizes the energy taken from the grid. This thesis proposes a linear optimisation model to optimise the available energy resources, within a LV distribution network, in such a way that congestion will be avoided and self-consumption will be increased. A focus is put on optimising the new Vechtrijk construction project. The microgrid designed for this new construction project includes EVs, a communal BESS and a communal PV system, while also incorporating grid services. Using MILP modelling techniques, the optimisation will be done using Pyomo, an optimisation modelling language within Python.

1.6. Thesis subdivision

Creating the design for a smart grid is divided into three subgroups/subprojects. The Topology subgroup (B.1) will develop a new design of a residential distribution network for the neighbourhood which includes the selection of components in terms of size and capacity. The main goal is to withstand congestion and integrate PV systems and electrical charging points for at least each resident. The Forecasting subgroup (B.2) will be making two forecasting models based on machine learning. The models will forecast the load and the PV power generation of the neighbourhood and can be used to determine whether congestion or voltage issues will occur. It will use old load data of several households (updated with EV charging and electric heating data), old PV generation data, old weather data and weather forecasts as inputs to forecast the load and generation demand. The optimisation Control subgroup (B.3) will create a controller for the grid by creating an optimisation function for the power flow. Examples of the decision variables are the rate of charge for the EV batteries and the battery storage system.

This report will provide an extensive explanation of the work done by subgroup B.3. First the program of requirements for the optimisation model will be presented. Then the design process will be explained. After this, the implementation process and validation tests will be presented and the final part is a discussion of the results.

2

Program of Requirements

This section will explain the requirements of the optimisation part of this project. Since the focus has been laid on the new construction in Weesp, namely the Vechtrijk new construction, the specifics of this neighbourhood will reflect in the program of requirements. These being that the neighbourhood consists of 37 households, divided into 4 categories: detached houses, semi-detached houses, terraced houses and apartments. For each of these categories the load profile will be predicted by the Forecasting subgroup (B.2). However since the projects run in parallel, the measurements from the databases are used for the model in substitution of the predictions. The same will hold for the weather predictions. These databases contain load and weather measurements from the Netherlands. The azimuth angle and tilt of the PV panels are out of the scope of this project and will thus not be taken into account. Finally it is assumed that each parking space in the neighbourhood is occupied by an EV. This accounts to a total of 54 EVs.

Another important aspect of the model is keeping the batteries in the grid at a safe SOC. An optimal depth of discharge (DoD) for a battery is 77% [24]. For ease of use, the DoD is incorporated into the model. This means that the capacity of the battery in the model is 23% smaller than that of a physical battery would be.

In this report, the requirements are developed with the MoSCoW method. In this method, each requirement is prioritized and divided into a specific section: ”Must have”, ”Should have”, ”Could have” and ”Will not have” [25]. This form of listing requirements gives everyone involved in a project a quick and simple view into top priorities and less prioritized parts of the projects and provides an easy way for outsiders to know the intention of the project.

To conclude, all requirements, while making use of the MoSCoW method, are presented below:

Must have:

1. The model must receive weather prediction per hour and interpolate this to intervals of 15 minutes.
2. The model must receive electricity usage prediction every 15 minutes.
3. The model must receive EV electricity demand prediction every 15 minutes.
4. The model must use as little energy from the external grid as possible.
5. The model must keep track of battery’s state of charge using predictions.
6. The model must keep the battery within a DoD of 77%.
7. The model must charge all EVs to 100% before each morning.

Should have:

1. The model should be able to shut curtail EV charging, as well as charge EVs at different power levels.
2. The model should be able to curtail solar energy if needed.
3. The model should minimize on system operational costs.

Could have:

1. The model could extract energy from the EVs to provide to the neighbourhood.
   (a) The model cannot discharge the EVs further when the SOC is under 20%.

Won't have:

1. The model won't perform any demand side management for the houses (e.g. delayed/planned consumption).
2. The model won't tweak energy usage in shorter intervals than 15 minutes.

3

Component Design

3.1. Modelling

In an effort to optimise the energy usage within the neighbourhood, first the inputs and outputs of the optimisation model must be clearly defined. The model must use these inputs to optimise the desired output. A black box model of the optimisation model can be seen in Figure 3.1.

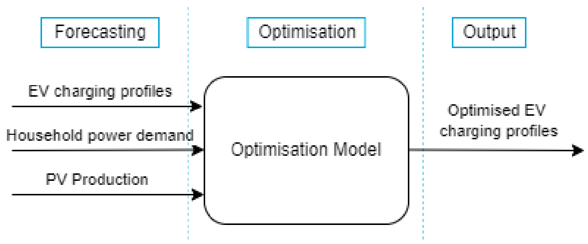


Figure 3.1: Black box model of the optimisation model showing the inputs and output

As seen in the black box model, the optimisation model gets the following predictions as inputs: PV power generation, household power demand and the EV charging profiles. These predictions are supplied to the model with a time horizon of five days. As mentioned previously in the program of requirements, these predictions would normally be provided by the Forecasting subgroup (B.2), however due to the parallel nature of the project, the data used for prediction is used as a substitute for the actual prediction. These inputs are used by the model to calculate the power generation of the PV, the total load and to know how to optimise the EV charging profiles. As for the output, the optimisation model provides the optimised EV charging profiles. The way these EV charging profiles are determined by the model is explained in Section 4.5. However, not only the optimised EV charging profiles can be extracted from the model. For instance the state of energy (SOE) of the BESS system or the generated PV power at a certain time could also be extracted from the model. While these are not necessarily outputs, they do provide the necessary information to inspect the model. These outputs will be discussed in Chapter 5.

3.1.1. Pyomo

In order to create the digital optimisation model, first a mathematical optimisation problem has been created. This is shown in Section 4.6. In order to convert this mathematical model into a digital model, the Pyomo environment within Python is used [26][27]. Pyomo is an open source software package for modeling and solving mathematical optimisation problems in Python. Pyomo allows for a large variety of optimisation techniques to be implemented. For this thesis, a LP model has been created since these models can be solved by commercially available solvers. Additionally, the speed at which the model can be solved is much higher than for other variants of optimisation models. This is however at the cost of a little bit of accuracy, but since large amounts of power are handled in this thesis, slight inaccuracies are negligible for the overall result. As for the solver, Gurobi has been chosen. This due to this optimisation solver being industry standard and being regarded as the fastest solver available [28].

3.2. Data analysis

The model of the neighbourhood network requires three sets of data. The power demand from the houses, the generated power from the PV system and the power demand for EV charging. This section will elaborate upon the contents and the retrieval of each of these datasets

3.2.1. Household power demand

The dataset of the household power demand was created from a combination of multiple datasets. First, a record of gas and electricity demand of 82 houses during the whole year of 2013 (in intervals of 15 minutes) was retrieved from Liander [29]. Then, only the clients with the correct house types and complete datasets were selected. The gas demand needed to be converted to electricity demand since the assumption was made that the houses will not have a gas connection. This conversion was performed and created by the Topology subgroup (B.1). A correct conversion rate was important since for example devices like heat pumps are more efficient than central heating boilers. According to [30] about 75% of the gas is being used for heating your house, 20 % of the gas is being used for heating water and the remaining 5 % is being used for cooking. Heating the house and water can be done using a heat pump with approximately a coefficient of performance (COP) of 3 [31]. This was used in Equation 3.1 to calculate 95% of the gas power converted to equivalent electrical power.

$$P_{el}[kW] = \frac{P_{gas}[kW] \cdot 0.95 \cdot \eta_{\text{boiler}}}{\text{COP}_{\text{heat pump}}} \tag{3.1}$$

The remaining 5% was converted using Equation 3.2.

$$P_{el}[kW] = \frac{P_{gas}[kW] \cdot 0.05 \cdot \eta_{\text{furnace}}}{\eta_{\text{induction}}} \tag{3.2}$$

where η_{furnace} = 85% and $\eta_{\text{induction}}$ = 35% [31].
The total demand per client was defined as the summation of the converted gas demand and original electricity demand.
Finally, an average was taken at each timestep for the different house types and a data profile was created for the Vechtrijk neighbourhood with a sum of 23x the average of the terraced houses, 11x the dataset of an apartment (since there was only one set available), twice the average of the semi-detached houses and one dataset of a detached house (since there was only one set available again). This thus created a dataset of the electricity demand of the entire neighbourhood for one year in intervals of 15 minutes.

3.2.2. PV power generation

The generated PV power was based on an open weather dataset from the KNMI of the Netherlands. This dataset contains the global radiation (in J/cm2) per hour of 2013. This dataset has been selected to match the same dates as the dataset for the demand from the houses. However this dataset was in time intervals of an hour, therefore it had to be linearly interpolated to be in line with the timestep of the model. The generated power was calculated by using the radiation, the total surface of the PV panels and the efficiency of the PV panels. First, the radiation had to be converted to W/m^2, which was then used to calculate the generated PV power:

$$P_{PV} = E_e * A_{PV} * \eta_{PV} \tag{3.3}$$

With $\eta_{PV} = 20\%$, since solar panel efficiencies are expected to achieve a maximum of 26% in the upcoming years [32]. However since the edges of the solar panels are not accounted for in the area estimation, the efficiency is chosen lower to compensate for this. The PV system would be placed on every roof in the neighbourhood and above the central parking lot. The measurement of this surface led to a total of 1178 solar panels, assuming a size of 1.7 m^2 per panel. The final result was a total PV power generation of the neighbourhood per 15 minutes for the year 2013.

3.2.3. EV power demand

For the EV charging demand, a dataset created by M. Muratori in [33] based on modeling reported in [34] was used. This dataset contains a power demand for charging the electric vehicles of 200 households in the US over one year in timesteps of 10 minutes. To use this data, a summation of power demand of 54 randomly determined clients have been selected to represent the neighbourhoods EV load. The 10 minute timestep was converted to a 15 minute timestep by removing one of 3 timesteps consistently. This method was effective and quick, but not very precise. But due to lack of time, the choice was made to process the data as mentioned.

3.3. Component sizing

A large amount of components in the model could be selected and sized freely. The amount of PV panels and EVs were selected by the whole project group since knowledge of each field in this project was necessary for this choice. The capacity of the BESS was selected by this subgroup, namely B.3, since the result graphs of this thesis show the clearest consequences of the BESS capacity. And finally, the transformer rate of power was selected by the Topology subgroup (B.1) and has a value of 630 kW.

3.3.1. PV: Number of PV panels

Since the Netherlands has plans to switch to almost 100% renewable energy by 2050 [35], the decision was made to place as many solar panels as possible in the neighbourhood. This means that all the roofs have solar panels, and a roof is placed above the central parking lot with additional solar panels. To find the amount of solar panels to be installed, the total roof surface has been measured and divided by solar panels of $1.7\,\mathrm{m}^2$ (since this is the average size of a solar panel in the Netherlands [36]). It was important to keep in mind that only a whole panel can be installed, and thus the amount of panels had to physically fit on the roof. Finally, this added up to 1178 solar panels.

3.3.2. EV: Number of EVs

Due to the rapidly increasing use of EVs, it was hard to predict how many EVs will be in use in the future. In 2019 the total amount of EVs in the world recorded a yearly increase of 40% [37]. Therefore to make the design future-proof, the assumption has been made that each parking spot will be occupied by one EV. As a result, the model contains 54 EVs in total. This corresponds to the total number of parking spots, both public and private as can be seen in Figure 1.1.

3.3.3. BESS: Capacity and maximum power

Sizing the BESS appropriately in reference to the model is a very important aspect since these systems are very costly and thus a huge waste of money and resources if not used to their full extent. In contrast, sizing the BESS too small can result in waste of renewable energy generation. With the PV system size already determined, as discussed in Subsection 3.3.1, the total PV generation can be calculated and consequently the excess generated power not used by the EVs or households can be calculated too. The Matlab code for these calculations is shown in Appendix A.1. Here, the excess generated power was converted to excess energy per day, since the BESS functions as an addition to the PV system which cycles per day. In Figure 3.2, the excess energy per day is shown. It is obvious that the excess generation varies drastically between winter and summer. Therefore the point at which there is a steady excess generation per day was observed. These dates are the 14th of April and the 11th of September as seen in Figure 3.2. These dates give insight in how much excess generation the model will generate when PV generation starts to exceed demand. Since the time horizon for our model is five days, the five days past the 14th of April and the five days preceding the 11th of September were considered. Over these weeks, the average excess energy generated was computed and is shown in Table 3.1.

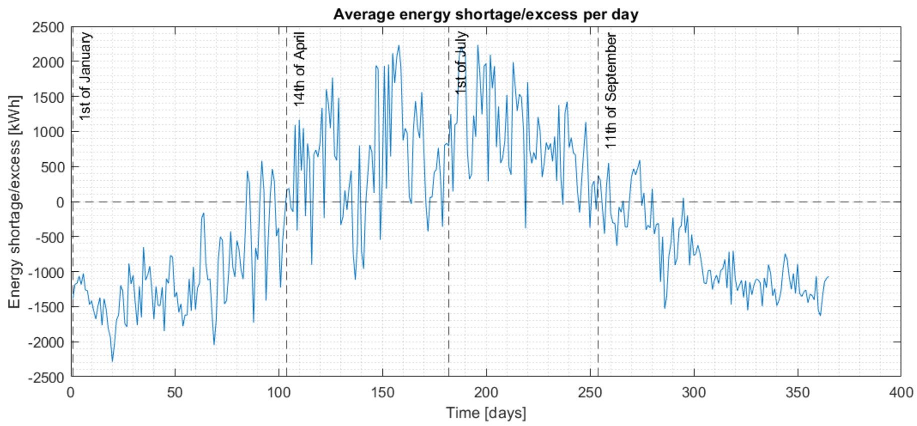


Figure 3.2: Average energy shortage/excess per day

As previously stated and seen in Figure 3.2, the values for the 14th of April and the 11th of September are at the edges of days where PV generation exceeds demand, therefore the BESS capacity should be at least as large as the excess energy generated in these weeks in April and September.

Table 3.1: The average shortage/excess of energy per day for a week

Time of year	Average excess energy [kWh]
January	-1180.8
April	280.0
July	1240.3
September	290.9

While the technological advancements in energy storage are improving greatly, it is unsure which capacity, rated power and efficiency these might achieve in the future. Therefore an already available BESS was selected. Looking at commercially available and popular BESSs, the Tesla Powerwall 2 was considered as a reference for our model. As specified by the company, this BESS is optionally expandable up to 10 systems stacked together [38]. Therefore it gives great opportunities to customize the BESS to the needs of any neighbourhood. In reference to the minimum capacity found in Table 3.1 and the DoD requirements stated in Chapter 2, 30 Powerwalls were selected as the BESS for the Vechtrijk neighbourhood. Such that the desired minimum capacity was reached while cutting down on total price. Looking at the datasheet of the Powerwall [39], gives us the following specifications for the BESS:

Table 3.2: BESS specifications

Rated Power	150 kW [39]
Capacity	405 kWh [39]
Efficiency	90 % [39]

However since the DoD used is 77%, the maximum capacity of the BESS within the model becomes:

$$E_{BESS,model} = E_{max} * DoD \implies E_{BESS,model} = 405 * 0.77 \approx 312\text{kWh} \tag{3.4}$$

4

The Optimisation Model

This chapter will elaborate upon the optimisation model. The Python code for this model can be found in Appendix A.2. The different components of the model will be explained in the same order as the code is written, namely first the parameters, then the variables and decision variables, and lastly the constraints. Each of these components are tied together in a Python Pyomo model. Details on the Pyomo method will be explained in the last section.

4.1. Sets

A set in the model is created to provide an index for parameters and variables [40]. The model for this project has 3 sets, one for time, one for the BESS and one for the transformer. The time set (set **T**) is used for every parameter and variable that has a different value for every timestep. This set is thus used very often. The set consists of all time slices with timesteps of 15 minutes for one year; $t \in \mathbf{T} : 1 \leq t \leq 35040$.
The set for the BESS and the transformer is created for a possible situation where our model is used for a neighbourhood with multiple BESSs and transformers. At the moment our model only has one battery and transformer, so the index is of little purpose but this way the setup is ready for expansion.

4.2. Parameters

The parameters in an optimisation model provide the input data that is necessary to solve the model. This model has constant parameters as well as parameters that change over time (set **T**). These multiple-valued parameters consist of the generation data and the load data of the consumers and the EV charging. The constant valued parameters consist of characteristics of different components in the network, such as the maximum SOE or efficiency of the BESS. The BESS parameters, as well as the transformer maximum power parameter only use the battery and transformer sets if multiple are present in the model. A parameter that does not use any set is the size of the timestep, which is 0.25 hour according to the program of requirements. A list of all of the parameters and so-called variables can be found in Table 4.1.

4.3. Decision variables

The decision variables are variables of which the model solver can change the values. Choosing the ideal values for the decision variables is essentially what solves optimisation problems. The decision variables are also the only values which the model solver can change. In short, this model contains a decision variable for the power flow and curtailment at each of the components of the model.
The BESS has five decision variables, namely, the SOE, the charging power, the discharging power and two binary categorical variables to decide whether the battery is charging, discharging or idle. The SOE is defined by the initial SOE, the efficiency and the charging and discharging energy.
The transformer has three decision variables, namely, the positive power flowing through the transformer, the negative power (this is fed back to the grid operator), and a binary categorical variable that ensures that the power flow at the transformer will always be either negative or positive.

The PV system only has two decision variables, the produced power and the curtailment rate variable. The EVs have five decision variables, of which one is the power supplied to the EVs and the other four collaborate in order to create a system where the EV charging can be curtailed to make the system more efficient. But since the vehicle's primary function is to drive, a limit of the amount of timesteps that the charging can be delayed has been added. This is why there are so many decision variables for this component. The implementation of this system is described in Section 4.5 Constraints.

Parameters	
δt	Timestep of the model [15 min]
P_{max}^{bat}	Rated power of the BESS [kW]
SOE_{max}^{bat}	Maximum state of energy of the BESS [kWh]
SOE_{ini}^{bat}	Initial state of energy of the BESS [kWh]
η^{bat}	Efficiency of the BESS
P_{max}^{TF}	Rated power of the neighbourhood transformer [kW]
P_t^{load}	Power demand from the houses at time t [kW]
$P_t^{PV,available}$	Available power from the PV system at time t [kW]
P_t^{EV}	Power demand from the EVs at time t [kW]
Decision Variables	
SOE_t^{bat}	State of energy of the battery at time t [kWh]
$P_t^{bat,ch}$	Charging power of the BESS at time t [kW]
$P_t^{bat,dis}$	Discharging power of the BESS at time t [kW]
u_t^{bat}	Binary variable indicating whether the BESS is charging or discharging at time t
$u_t^{bat,idle}$	Binary variable indicating whether the BESS is idle or charge/discharge at time t
P_t^{grid+}	Positive power at the transformer (demand from the neighbourhood) at time t [kW]
P_t^{grid-}	Negative power at the transformer (supply from the neighbourhood) at time t [kW]
u_t^{grid}	Binary variable indicating whether the neighbourhood is supplying or demanding power from the external grid at time t
$P_t^{PV,produced}$	Power generated by the PV system at time t [kW]
u_t^{PV}	Variable between 0 and 1 that curtails the amount of power generated by the PV system at time t
$P_t^{EV,delay}$	Delayed/unsupplied charging power from the EV power demand at time t [kW]
$P_t^{EV,total}$	Sum of the EV power demand at time t and delayed EV power at time t-1 [kW]
$P_t^{EV,sup}$	Supplied charging power of the EVs at time t [kW]
u_t^{EV}	Variable between 0 and 1 that curtails the amount of charging power supplied to the EVs at time t

Table 4.1: Overview of the parameters and decision variables of the optimisation model

4.4. Objective function

The objective function of the optimisation model defines the goal of the optimisation. The goal is to either minimize or maximize the output value of this function. In the case of the network controller, the objective function is minimized and consists of the total power at the transformer. This being the sum of the positive and negative power at the transformer that was mentioned in Section 4.3. In other words, the total energy that is taken from the grid as well as supplied back to the grid at the end of the time horizon is minimized. Saving as much energy as possible prevents congestion at the transformer side, creates more self-sufficiency and also saves on electricity costs for the neighbourhood. Since the goal of this project is to create a smart and sustainable distribution network which promotes self-sufficiency, it is important to look at the energy consumption and not the costs. A mathematical description of the objective function is given in Equation 4.1.

4.5. Constraints

In simple words, the constraints of the model define the rules that the model solver has to follow when trying to minimize the objective function.The combination of the constraints and the decision variables are in a way the description the functions in the neighbourhood network. Below, a description will be given of the constraints in the model. An overview the constraints in mathematical form is given in Section 4.6.

The BESS is defined by the constraints that describe the current SOE (Eq. 4.2, 4.3), the limits of the SOE (Eq. 4.4), the charging and discharging power (Eq. 4.5, 4.6) and the state constraint (Eq. 4.7). The current SOE constraint consists of a summation of the previous SOE, the added charging power and subtracted discharging power (while holding efficiency in account). The limits of the SOE constraint keeps the SOE between safe charging limits as mentioned in Ch. 2. The charging and discharging power constraints define these powers as either a negative or positive value of the battery's rated power depending on the state of the BESS. And lastly, the state constraint ensures that the BESS is always operating in only one mode at the same time; charging, discharging or idle.

The first transformer constraint defines the total power at the transformer to be a summation of all the supplies and demands in the neighbourhood network (Eq. 4.17). The second and third constraints (Eq. 4.18, 4.19) define the limits of the positive and negative power that the transformer can handle.

The PV system is fairly simple and has only two constraints. One for defining a curtailing variable between 0 and 1 (Eq. 4.8), and one for defining the produced power by the PV panels after curtailment (Eq. 4.9).

The EV charging system is the most intricate system in our model. In order to curtail the EV charging, a large amount of constraints are necessary. First, there are constraints that define curtailment variable between 0 and 1 (Eq. 4.11) and make sure that there is no curtailment at the end of the total model time (Eq. 4.10). The latter is to prevent the model from simply disregarding power that was demanded by the EV demand data (at the end of the models timeline). The next sets of constraints (Eg. 4.12, 4.13 define the total demand of power for the EVs as a summation of the power delayed/unsupplied power at the previous timestep (if it exists) and the EV power demand of the current timestep. The delayed power is in turn the difference between the original power demand and the supplied power to EVs at the current timestep (Eq. 4.14. And finally the supplied power is the previously mentioned total power EV demand multiplied with the curtailing variable (Eq. 4.15). The very last part of the EV constraints creates a limit on the amount of time that the EV demand can be delayed. This is defined by Eq. 4.16 with a summation of all of the supplied EV power until the current timestep that has to be larger than the summation of the original demand until 8 timesteps before. This thus limits the delay to 8 timesteps which equals to two hours. This limit was created to make sure that curtailing the charging will affect the user experience of the vehicles as little as possible.

4.6. Mathematical model

Objective function:

$$ObjFnc = minimise\left[\delta t \cdot \sum_{t} P_t^{grid+} + P_t^{grid-}\right] \tag{4.1}$$

BESS constraints:

$$SOE_t^{bat} = SOE_{ini}^{bat} + \delta t \cdot (\eta^{bat} P_t^{bat,ch} - \frac{P_t^{bat,dis}}{\eta^{bat}}) \qquad t = 1 \tag{4.2}$$

$$SOE_t^{bat} = SOE_{t-1}^{bat} + \delta t \cdot (\eta^{bat} P_t^{bat,ch} - \frac{P_t^{bat,dis}}{\eta^{bat}}) \qquad \forall\ t > 1 \tag{4.3}$$

$$0 \leq SOE_t^{bat} \leq SOE_{max}^{bat} \qquad \forall\ t \in \mathbf{T} \tag{4.4}$$

$$P_t^{bat,ch} \leq P_{max}^{bat}(u_t^{bat}) \qquad \forall\ t \in \mathbf{T} \tag{4.5}$$

$$P_t^{bat,dis} \leq P^{bat,max}(1 - u_t^{bat}) \qquad \forall\ t \in \mathbf{T} \tag{4.6}$$

$$0 \leq u_t^{bat} + u_t^{bat,idle} < 1 \qquad \forall\ t \in \mathbf{T} \tag{4.7}$$

Photovoltaic panel constraints:

$$0 \leq u_t^{PV} \leq 1 \qquad \forall\ t \in \mathbf{T} \tag{4.8}$$

$$P_t^{PV,produced} = P_t^{PV,available} \cdot u_t^{PV} \qquad \forall\ t \in \mathbf{T} \tag{4.9}$$

Electric vehicle constraints:

$$u_t^{EV} = 1 \qquad t = length(\mathbf{T}) \tag{4.10}$$

$$0 \leq u_t^{EV} \leq 1 \qquad t < length(\mathbf{T}) \tag{4.11}$$

$$P_t^{EV,total} = P_t^{EV} \qquad t = 1 \tag{4.12}$$

$$P_t^{EV,total} = P_t^{EV} + P_{t-1}^{EV,delay} \qquad t > 1 \tag{4.13}$$

$$P_t^{EV,delay} = P_t^{EV,total} - P_t^{EV,sup} \qquad t \in \mathbf{T} \tag{4.14}$$

$$P_t^{EV,sup} = P_t^{EV,total} \cdot u_t^{EV} \qquad t \in \mathbf{T} \tag{4.15}$$

$$\sum_{i=0}^{t-8} P_i^{EV} \leq \sum_{i=0}^{t} P_i^{EV,sup} \qquad t > 8 \tag{4.16}$$

Transformer constraints:

$$P_t^{grid+} - P_t^{grid-} = P_t^{load} - P_t^{PV,produced} - P_t^{bat,dis} + P_t^{bat,ch} + P_t^{EV,sup} \forall\ t \in \mathbf{T} \tag{4.17}$$

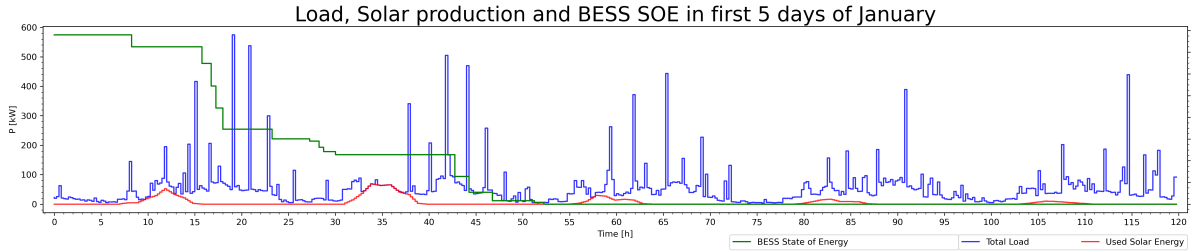
$$P_t^{grid+} \leq P_{max}^{TF} \cdot u_t^{grid} \qquad \forall\ t \in \mathbf{T} \tag{4.18}$$

$$P_t^{grid-} \leq P_{max}^{TF} \cdot (1 - u_t^{grid}) \qquad \forall\ t \in \mathbf{T} \tag{4.19}$$

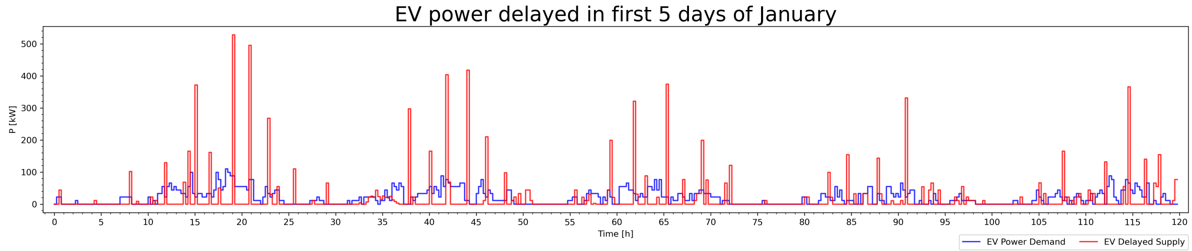
5

Results and Discussion

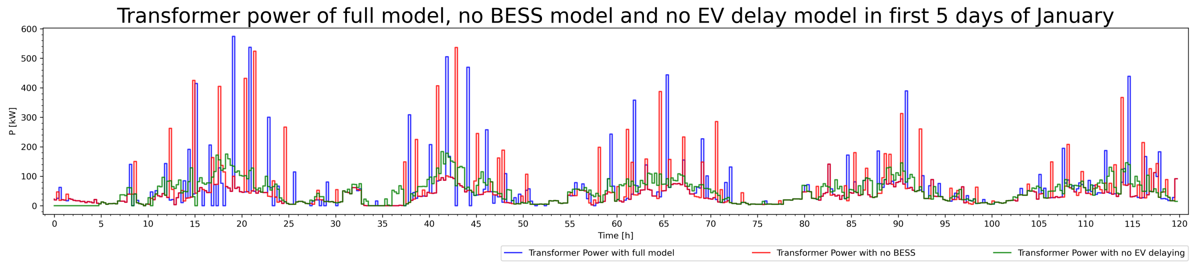
This chapter will present and discuss the results of the optimisation model. The results during three seasons will be discussed, namely during the winter, spring and summer (in the same order). For each result, the model is executed over a timelength of 5 days.
The first winter figure is Figure 5.1. The graphs in this figure show: the power balance in (a), EV delaying in (b) and the transformer power of different models in (c). Firstly, when looking at the power balance of the optimisation model in January, what becomes clear is that even though the sizing of the PV system, as mentioned in subsection 3.3.1, is quite large, there still is not enough energy generated during winter. This is reflected in the SOE of the BESS. Large peaks in load and small peaks in PV generation as seen in Figure 5.1a means that there is no excess energy to charge the communal battery. The lack of charging power is shown very clearly in the fact that the SOE of the battery only decreases in time. The energy shortage is also apparent when observing the winter months in Figure 3.2 and Table 3.1. This in turn makes the BESS system have little effect on the power at the transformer side. This is confirmed by the small difference in peak heights between the blue and the red line in Figure 5.1c (namely the power at the transformer with and without the BESS). The only significant difference between the two lines is the timing: some of the peaks are shifted in time due to the optimised EV delaying. As seen in Figure 5.1b, the supplied power (red) has peaks mostly at intervals of 2 hours. This interval corresponds exactly with the maximum delay window of the EVs. In Table 5.1 the effects of changing the EV delaying window are shown. Unfortunately, during winter the effect of EV delaying is almost negligible since the amount of energy saved by EV delaying already stagnates at a delay window of 1 hour and only saves 3.49 kWh in 5 days. This is expected however, since having not enough energy stored in the BESS and having not enough energy production from the solar panels, the power demand must come from the transformer anyway.



(a)



(b)



(c)

Figure 5.1: Model characteristics during winter

Table 5.1: Effect of different EV delaying windows on the energy needed from the transformer over five days during the winter

EV delay window [hours]	Energy from transformer [kWh]	Energy saved compared to 0 EV delay [kWh]	Energy saved compared to 0 EV delay [%]
0	5840.26	0	0
0.5	5837.84	2.42	0.04
1	5836.77	3.49	0.06
1.5	5836.77	3.49	0.06
2	5836.77	3.49	0.06
2.5	5836.77	3.49	0.06
3	5836.77	3.49	0.06
3.5	5836.77	3.49	0.06

Figure 5.2 shows that during these five days in January there is no PV curtailment since the total energy is the same height as the used energy every day. Looking at Figure 5.1a, one can see the SOE of the BESS declining from its original SOE towards zero without charging once. These two events must mean that all the PV energy is going straight to the houses and EVs to reduce the objective function, i.e. reduce the power needed from the external grid. Therefore the BESS is not charged during these 5 days in January.

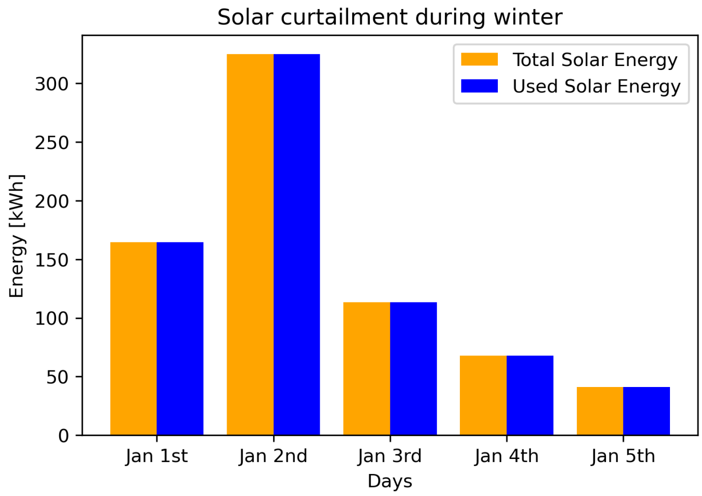


Figure 5.2: The total and used solar energy during winter

Similarly, the system has also been simulated during summer, which can be seen in Figure 5.3. In contrast to the model during winter, the model has an excess of energy production during summer, which is shown by the high peaks in PV production compared to the load in Figure 5.3a. Each day portrays a clear pattern of high PV production and low load during the day and low PV production and high load in the evening. The SOE of the battery follows this pattern nicely by charging when there is more production than load and discharging when it is the other way around. Here, the SOE of the BESS is at its maximum storage value at certain periods during daylight which is why it looks like the SOE curve is clipped at 312 kWh. During these periods, PV curtailing is taking place since the transformer power, shown in Figure 5.3c, is kept zero and the PV production is exactly matched with the load during these same periods. This proves that the model also minimises the amount of power fed back to the external grid. Figure 5.4 shows the amount of PV energy that is produced per day. The yellow bars are the amount of energy available each day, and the blue bars are the amounts that are generated. The difference between the two is thus the amount of curtailed energy. Here it is evident that a large part of the PV system is curtailed throughout the day during summer, since some days only about a third of the total available solar energy is used. One might suggest downsizing the PV system as currently it does not operate at its full potential during the summer and thus could be considered as a waste of materials and/or money. This is however not desired, since this will create even larger problems during periods when solar irradiance is low. A better solution would be to introduce a seasonal BESS which stores most of the excess generated solar energy to be used later when energy production is low (i.e. during winter). Another solution would be to introduce a different kind of renewable energy source which is not seasonally bound the same way as PV systems are. Introducing a different kind of renewable energy source would in turn give opportunity to downsize the PV system if deemed necessary.

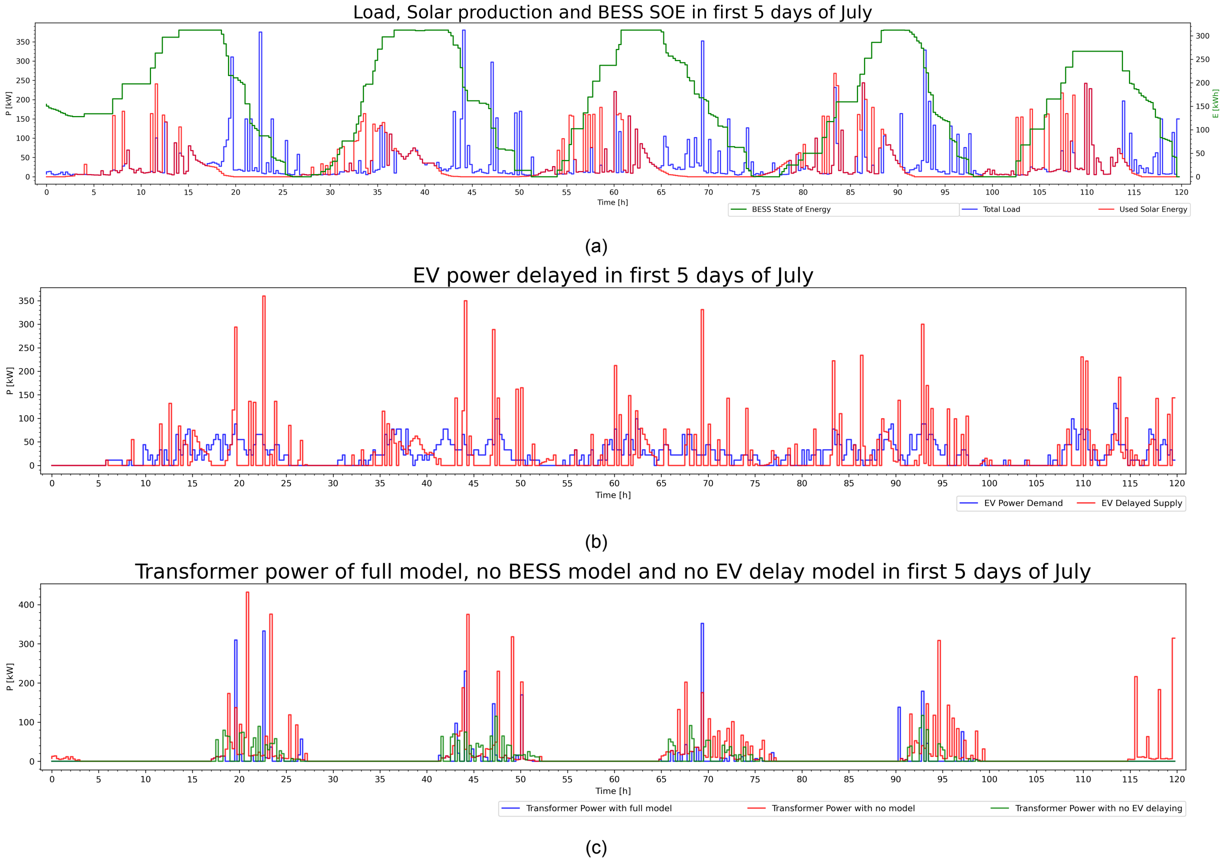


Figure 5.3: Model characteristics during summer

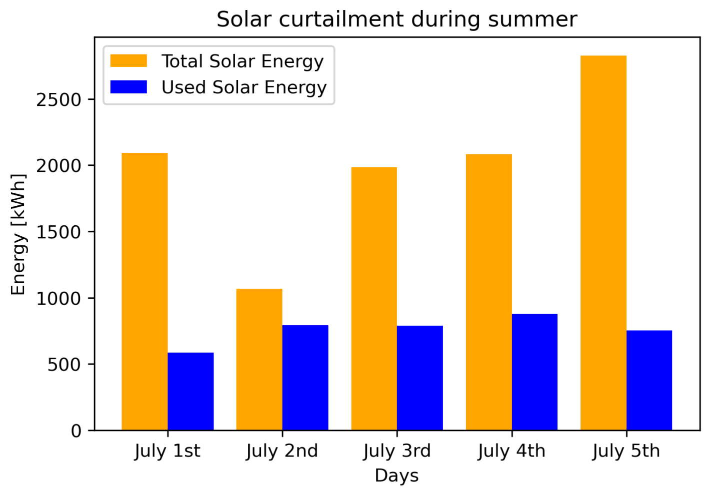


Figure 5.4: The total and used solar energy during summer

As for the EV delaying during summer shown in Figure 5.3b, the effect it has on the transformer power demand shown in Figure 5.3c is substantially larger than during winter. There are fewer peaks when looking at the power at the transformer with the full model (blue) in Figure 5.3c, compared to the delayed EV supply (red) in Figure 5.3b. For example, between hours 40 and 50, where there is little to no PV generation present, the transformer power only shows three peaks, whereas the delayed EV supplied power shows more during this same period. This shows that delaying the EV charging over a window results in less power at the transformer. Since the model has optimised the EV charging within its

2 hour window, there are still some peaks visible due to the fact these loads cannot be delayed any longer than 2 hours. Therefore these loads have to be supplied at a suboptimal moment. This is also reflected in Table 5.2, where a substantial amount of energy can be saved by delaying the charging of the EVs over a larger window. A window of 1 hour already saves 8.43 kWh over 5 days, which is more than double the amount that can be saved by EV delaying in the winter (namely 3.49 kWh). Enlarging the charging window gives the model more freedom into when to charge the vehicles within that window. Combining this with an energy generation excess at certain times, it becomes evident that shifting the EV charging to a time when generation is high, reduces the energy demand from the transformer. Table 5.2 shows that for a delay window between 0 and 2.5 hours, the improvement of energy consumption in percentage doubles every step (of 0.5 hours). This leads to almost 9% less energy consumption over 5 days with a delaying window of 2.5 hours. When increasing the delay window even more, the improvement in energy consumption starts to level off gradually. A delay window of 3.5 hours reduces the energy consumption of the neighbourhood over 5 days by 17% (which equals 126.13 kWh). To keep the delaying feature user-friendly, the choice was made to delay up to 2 hours. This was the largest time frame that was expected to have little effect on supplying the consumers with enough charge for their vehicles. In a real project, the cost of these energy savings should be compared to the cost of living and user friendliness to choose the optimal value for the EV delaying window.

Table 5.2: Effect of different EV delaying windows on the energy needed from the transformer over five days during the summer

EV delay window [hours]	Energy from transformer [kWh]	Energy saved compared to 0 EV delay [kWh]	Energy saved compared to 0 EV delay [%]
0	758.5	0	0
0.5	756.09	2.41	0.32
1	750.07	8.43	1.11
1.5	740.01	18.49	2.44
2	722.16	36.34	4.79
2.5	692.16	66.34	8.75
3	655.08	103.42	13.63
3.5	632.37	126.13	16.63

Finally, the model has also been tested during spring. This time, the model has a better balance between the total available solar energy and the used solar energy, which is shown in Figure 5.5. Just as in the previous figures, the amount of energy that is curtailed is the difference between the total solar energy and the used solar energy. While there is still curtailment on days where the solar irradiance is higher (for example April first, where about half of the total solar energy is used), the curtailment is visibly lower than that during the summer (where often only a third of the energy is used). This curtailment on days where the solar irradiance is higher is visible in Figure 5.6a. Here the used solar energy (red), sometimes exactly mirrors the load (blue), which means the model curtails the excess power generation in an effort to keep the transformer power at zero. This excess power generation could be stored in a season storage for more renewable energy usage. Whilst the excess generation could be stored, it means the seasonal storage needs to be of vast capacity if the storage already starts in April and last throughout the whole summer. This means that the initial decision of preparing for any demand increases has resulted in over-sizing the PV system for now, since is unsure by how much the energy demand will increase in the future. If the energy demand increases rapidly in the future, the large PV system might still be appropriate, however this is beyond the scope of this project.
The results from the model can be seen in Figure 5.6. When looking at the BESS SOE (green) in Figure 5.6a, it mimics the behaviour as seen in Figure 5.3a where the SOE rises during PV generation in the mornings and falls during the larger load in the evenings, apart from day three. On day three the solar generation is too little to have an impact on the model. This resembles the behaviour as seen in Figure 5.1, where the BESS is not charged due to too little irradiance during that day. Figure 5.6c clearly shows that the power at the transformer is clearly higher at the third day compared to other days because of the low irradiance and SOE of the BESS.

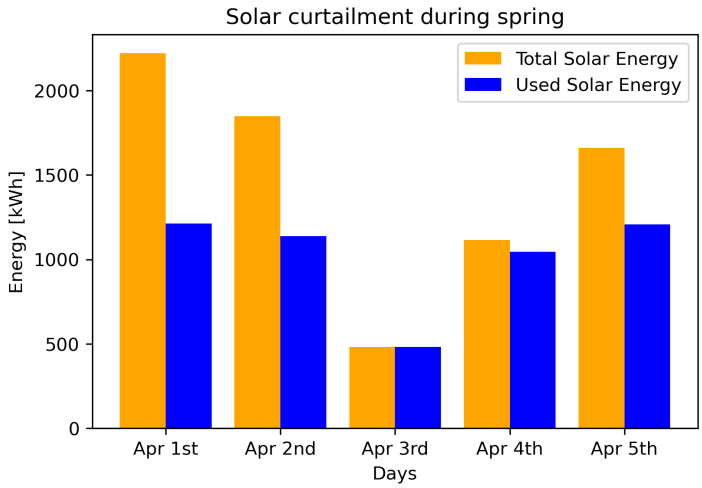
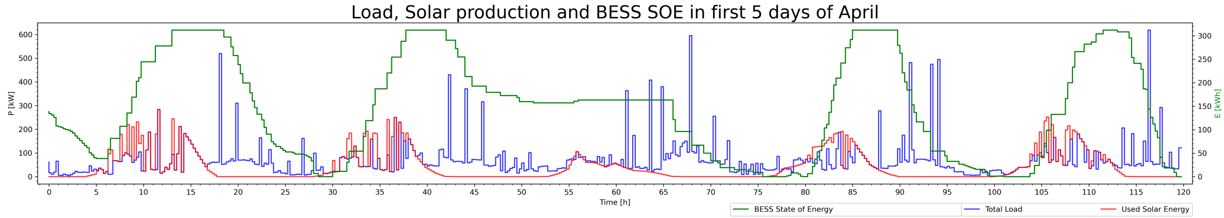
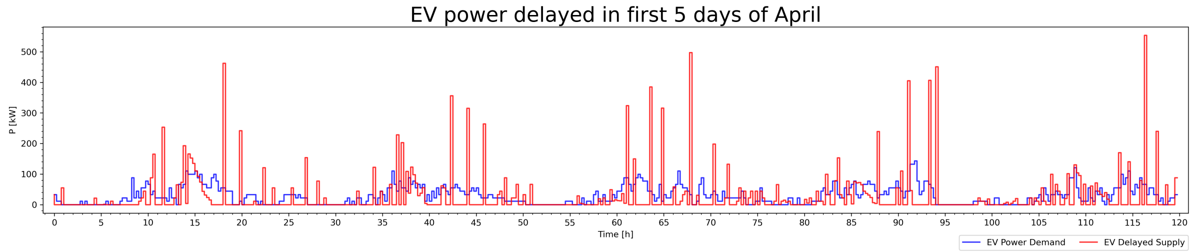


Figure 5.5: The total and used solar energy during spring

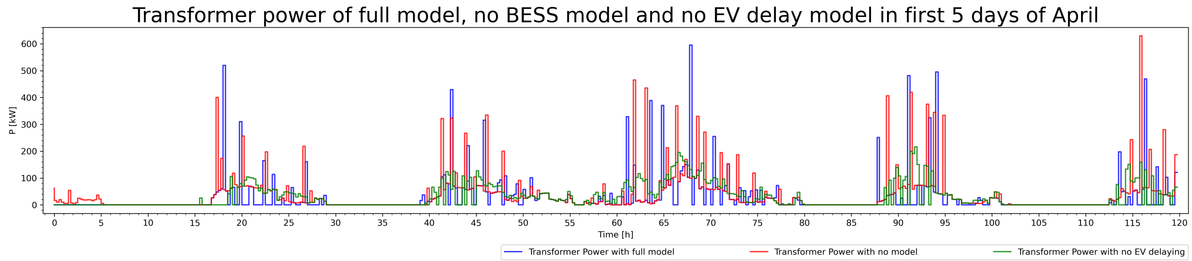
The EV delaying (red) in Figure 5.6b, has a reasonable impact on the power demand from the transformer (blue), as seen in Figure 5.6c. When looking between hours 90 and 95, three high peaks are visible in the load, as seen in Figure 5.6a. These peaks originate from the EV delaying, as these peaks are also present in Figure 5.6b. However in Figure 5.6c, while the three peaks are still present, the second peak has greatly reduced in size when looking at the transformer power with the full model (blue). This amidst a period where little to no PV generation is taking place. The effects of EV delaying becomes even more clear when inspecting Table 5.3, where extending the EV charging delay window shows a decrease in energy demand from the transformer. While there is a decrease (29 kWh saved with a 2-hour delay window), it is a little less than the decrease seen during summer (36 kWh saved with a 2-hour delay window), this is not surprising since both periods have shown excess PV energy generation. However, as seen in Figure 5.5, on the third day there is very little solar energy generation. This can also be seen in Figure 5.6c, where on the third day the demand from the transformer is highest. This explains why the EV delaying has less impact than during the summer. This also emphasizes how weather dependent the efficiency of the model is when the only generation is through solar power.



(a)



(b)



(c)

Figure 5.6: Model characteristics during spring

Table 5.3: Effect of different EV delaying windows on the energy needed from the transformer over 5 days during the spring

EV delay window [hours]	Energy from transformer [kWh]	Energy saved compared to 0 EV delay [kWh]	Energy saved compared to 0 EV delay [%]
0	3593.42	0	0
0.5	3592.89	0.53	0.01
1	3584.28	9.14	0.25
1.5	3578.55	14.87	0.41
2	3564.27	29.15	0.81
2.5	3556.56	36.86	1.03
3	3541.17	52.25	1.45
3.5	3524.34	69.08	1.92

To conclude, the effect the optimisation model has on the demand from the transformer is shown below in Table 5.4. Here, the total power demand from the EVs and households during the certain period is added and converted to energy and compared to the energy demand from the transformer with the optimisation model active. The result thus shows the effect of the communal battery, the PV panels and the optimisation model on the consumption of energy from the external grid. The improvement of the model on the energy consumption of the neighbourhood is calculated as the energy demand without the model subtracted with the energy demand with the model and the initial state of energy of the battery (namely 156 kWh). It is clear that the model improves the neighbourhood's energy consumption substantially in the spring and summer (namely 56% and 80%) but less in the winter (namely 10%). Since there is very little PV generation during the winter, there is no excess power to charge the communal battery. Thus, the model has much less room to optimize power consumption during the winter. But the large improvement during the summer proves that the smart distribution network designed in this project has a very positive effect on decreasing congestion and energy consumption.

Table 5.4: Effect of the optimisation model on the energy demand from the transformer

Time of year	Energy demand no model [kWh]	Energy demand with model [kWh]	Improvement [kWh] (excluding SOE_{ini}^{bat})	Improvement [%] (excluding SOE_{ini}^{bat})
winter	6689.21	5836.77	696.44	10.41
spring	8529.74	3564.27	4809.47	56.38
summer	4372.77	722.16	3494.61	79.92

6 Conclusion

The final form of the model met almost all of the requirements mentioned in Chapter 2. All of the ”must have” requirements have been met, two of the three ”should have” requirements were met and the ”could have” requirements were not implemented. The model saves up to 5 MWh of energy in one week for the Vechtrijk neighbourhood.

The first three ”must have” requirements were about the three input data types. These were indeed adapted and implemented in 15 minute intervals as explained in Section 3.2. The fourth requirement mentions that the model must use as little energy from the external grid as possible. This was achieved by defining the objective function to minimize the total power at the transformer as explained in Section 4.4. To minimize congestion in the Netherlands even more, the objective function also minimizes the power supplied back to the external grid. The fifth and sixth requirements state that the SOC of the BESS has to be tracked and kept between a safe depth of discharge. This is all implemented within the constraints of the BESS model (see Eq. 4.2, 4.3, 4.4). The final ”must have” requirement was that the EVs have to be fully charged before each morning. This was implemented by creating a limit to the amount of time that the EV demand can be delayed. This limit was set to 2 hours. That means that after someone plugs in their vehicle, it will only take up to 2 hours extra to reach a full charge. And since this limit is always present, this implementation also improves the user experience of the charging stations during the day.
The two ”should have” requirements that were successfully implemented were the curtailment of EV charging as well as the PV system. Both could be curtailed with all values between 0 and 1. The requirement that states that the model should optimise electricity cost was not implemented, since the decision was made that solving congestion was more important than saving money since congestion is a much larger problem.
The final requirement was a ”could have” requirement and stated that the model could extract energy from the EVs. This temporary storage solution was desirable to implement but very complicated to implement with the current EV input data, namely the charging power at every timestep. Therefore, this has been added as a suggestion for future work.

6.1. Future work

The results from the model were satisfactory with regard to the requirements. Nevertheless, there were points on which the model can be improved. This leads to the following possibilities to expand this work in the future:

- A seasonal BESS could be added to the model which should improve the performance during winter and reduce the curtailment during summer. This BESS would have to have a significantly large capacity and thus additional research should also concentrate on investigating the best material for storing such large amounts.
- Different kinds of renewable energy sources could be explored to have less reliance on solar irradiation.

- The EVs could also be added to the model as a temporary storage solution. If the model can extract power from the EVs, it is important to add a constraint that the SOC must not fall below a value where the consumers cannot drive their cars anymore. The model will also need data that contains the SOC of the vehicles upon arrival to implement an EV discharging feature.
- Peak shaving could be explored to lessen the unpredictability of peaks in demand from the transformer. Since these peaks may not be desirable for the power supplying companies.
- Delaying or controlling the usage of controllable devices (such as washing machines) could be introduced in the system to give more opportunities to optimise the power flow.
- In future work, this project will incorporate the forecasting data from group B.2 to investigate the effects of forecasting data on the powerflow.
- The timestep, currently 15 minutes, could be reduced to increase the effect the optimisation has on the system.
- An additional optimisation problem could be created to find the ideal size for the PV system and BESS in terms of cost and capacity.

Bibliography

[1] P. R. et al., "Greenhouse gas emissions in the netherlands 1990–2019, national inventory report 2021," 2021. `https://www.rivm.nl/bibliotheek/rapporten/2021-0007.pdf`, accessed on: 29/04/2022. Doi: 10.21945/RIVM-2021-0007.

[2] N. Brannan, "Liander niet goed voorbereid op overvol stroomnet, gemeente overvallen door problemen," 2022. `shorturl.at/ruzH9`, accessed on: 18/05/2022.

[3] M. J. L. et al., "Hernieuwbare energie in nederland - buitenluchtwarmte," 2021. `https://www.cbs.nl/nl-nl/longread/aanvullende-statistische-diensten/2021/hernieuwbare-energie-in-nederland-2020?onepage=true#c-7--Buitenluchtwarmte`, accessed on 29/04/2022.

[4] C. B. voor de Statistiek (CBS), "Aantal stekkerauto's, 2014-2021," 2021. `https://www.cbs.nl/nl-nl/maatwerk/2021/41/aantal-stekkerauto-s-2014-2021`, accessed on: 29/04/2022.

[5] "Aanbod vechtrijk xb2." `https://account.woneninweespersluis.nl/aanbod-vechtrijk/`.

[6] "Netcapaciteit en netcongestie." `https://www.rvo.nl/onderwerpen/zonne-energie/netcapaciteit-netcongestie`.

[7] "Capaciteitskaart nederland." `https://capaciteitskaart.netbeheernederland.nl/?_ga=2.56067567.538483043.1651476493-1899990241.1649077574`.

[8] U. Datta, A. Kalam, and J. Shi, "Smart control of bess in pv integrated ev charging station for reducing transformer overloading and providing battery-to-grid service," *Journal of Energy Storage*, vol. 28, 2020.

[9] R. H. Byrne, T. A. Nguyen, D. A. Copp, B. R. Chalamala, and I. Gyuk, "Energy management and optimization methods for grid energy storage systems," *IEEE Access*, vol. 6, pp. 13231–13260, 2018.

[10] M. T. Lawder, B. Suthar, P. W. C. Northrop, S. De, C. M. Hoff, O. Leitermann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, "Battery energy storage system (bess) and battery management system (bms) for grid-scale applications," *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, 2014.

[11] T. Wang, H. Kamath, and S. Willard, "Control and optimization of grid-tied photovoltaic storage systems using model predictive control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 1010–1017, 2014.

[12] W. Kempton and S. E. Letendre, "Electric vehicles as a new power source for electric utilities," *Transportation Research Part D: Transport and Environment*, vol. 2, no. 3, pp. 157–175, 1997.

[13] C. Le Floch, F. di Meglio, and S. Moura, "Optimal charging of vehicle-to-grid fleets via pde aggregation techniques," pp. 3285–3291, 2015.

[14] R. Alvaro, J. González, C. Gamallo, J. Fraile-Ardanuy, and D. L. Knapen, "Vehicle to vehicle energy exchange in smart grid applications," pp. 178–184, 2014.

[15] T. J. C. Sousa, V. Monteiro, J. C. A. Fernandes, C. Couto, A. A. N. Meléndez, and J. L. Afonso, "New perspectives for vehicle-to-vehicle (v2v) power transfer," 2018.

[16] K. Perez, M. Baldea, T. Edgar, G. Hoogsteen, R. van Leeuwen, T. van der Klauw, B. Homan, J. Fink, and G. Smit, “Soft-islanding a group of houses through scheduling of chp, pv and storage,” pp. 1–6, 2016.

[17] A. Molderink, V. Bakker, M. G. Bosman, J. L. Hurink, and G. J. Smit, “Domestic energy management methodology for optimizing efficiency in smart grids,” *2009 IEEE Bucharest PowerTech*, pp. 1–7, 2009.

[18] P. Huang and Y. Sun, “A robust control of nzebs for performance optimization at cluster level under demand prediction uncertainty,” *Renewable Energy*, vol. 134, pp. 215–227, 2019.

[19] P. Huang, H. Wu, G. Huang, and Y. Sun, “A top-down control method of nzebs for performance optimization at nzeb-cluster-level,” *Energy*, vol. 159, 2018.

[20] S. Walker, V. Bergkamp, D. Yang, T. van Goch, K. Katic, and W. Zeiler, “Improving energy self-sufficiency of a renovated residential neighborhood with heat pumps by analyzing smart meter data,” *Energy*, vol. 229, 2021.

[21] D. Asija and P. Choudekar, “Congestion management using multi-objective hybrid de-pso optimization with solar-ess based distributed generation in deregulated power market,” *Renewable Energy Focus*, vol. 36, 2021.

[22] J. A. Pinzon, P. P. Vergara, L. C. P. da Silva, and M. J. Rider, “Optimal management of energy consumption and comfort for smart buildings operating in a microgrid,” *IEEE Transactions on Smart Grid*, vol. 10, no. 3, 2019.

[23] N. K. Panda and N. G. Paterakis, “A multi-objective optimization model for the quantification of flexibility in a large business park,” 2021.

[24] L. Setyawan, J. Xiao, and P. Wang, “Optimal depth-of-discharge range and capacity settings for battery energy storage in microgrid operation,” pp. 1–7, 2017.

[25] K. Brush, “Moscow method.” `https://www.techtarget.com/searchsoftwarequality/definition/MoSCoW-method`.

[26] W. E. Hart, J.-P. Watson, and D. L. Woodruff, “Pyomo: modeling and solving mathematical programs in python,” *Mathematical Programming Computation*, vol. 3, 2011.

[27] M. Bynum, G. Hackebeil, W. Hart, C. Laird, B. Nicholson, J. Siirola, J. P. Watson, and D. Woodruff, *Pyomo - Optimization modeling in Python*, vol. 67 of *Springer Optimization and its applications*. Springer, 2021.

[28] Gurobi, “Gurobi optimization,” `https://www.gurobi.com/products/gurobi-optimizer/`, accessed on 3/06/2022.

[29] Liander, “Beschikbare data.” `https://www.liander.nl/partners/datadiensten/open-data/data`.

[30] V. de Vastelastenbond, “Gasverbruik: Gasverbruik in jouw huishouden,” 2020. `https://www.vastelastenbond.nl/energie/gasverbruik-in-uw-huishouden/`, accessed on: 31/05/2022.

[31] R. Khezri, A. Mahmoudi, and D. Whaley, “Optimal sizing and comparative analysis of rooftop pv and battery for grid-connected households with all-electric and gas-electricity utility,” *Energy*, vol. 251, p. 123876, 2022.

[32] “Solar panel effiency continues to improve as costs fall,” 2022. `https://www.tno.nl/en/focus-areas/energy-transition/roadmaps/renewable-electricity/solar-energy/solar-panel-efficiency/`, Accessed on 02/06/22.

[33] M. Muratori, “Impact of uncoordinated plug-in electric vehicle charging on residential power demand - supplementary data,” 2017.

[34] M. Muratori, M. J. Moran, E. Serra, and G. Rizzoni, “Highly-resolved modeling of personal transportation energy consumption in the united states,” *Energy*, vol. 58, 2013.

[35] “”central government encourages sustainable energy”.” ”https://www.government.nl/topics/renewable-energy/central-government-encourages-sustainable-energy”.

[36] “”afmeting van zonnepanelen, hoe groot zijn ze?”.” ”https://www.zonnepanelen.net/afmeting/, Accessed on 24/05/22”.

[37] IEA, “Global ev outlook 2020,” 2020. https://www.iea.org/reports/global-ev-outlook-2020, accessed on 6/06/2022.

[38] Tesla, “Powerwall,” https://www.tesla.com/powerwall, accessed on 5/06/2022.

[39] “Powerwall 2 datasheet,” 2019. https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall_2_AC_Datasheet_EN_NA.pdf, Accessed on 05/06/22.

[40] 2017. ”https://pyomo.readthedocs.io/en/stable/pyomo_modeling_components/Sets.html”.

A
Appendix

A.1. Design

```
area = 2002.6;      % PV system area
eff = 0.2;          % PV system efficiency

% get irradiance per hour
weerdata_2013 = readmatrix('weerdata_2013.csv');
data.solar = weerdata_2013(:,2);
data.hour = weerdata_2013(:,1);
data.solarpower_h = data.solar*area*eff/1000; %power in kW

% interpolate every 15 minutes
time = [1:1:8760];
interpolated = [1:0.25:8760];
data.solarpower_15 = interp1(time,data.solarpower_h,interpolated);
data.solarpower_15 = data.solarpower_15(:);     % convert to column vector

% write weather data back to excel in sheet 10
writematrix(data.solarpower_15,'variables_BAP_local.xlsx','Sheet',10);

% Get load data per hour
loaddata = readmatrix('latest_user_data.csv');
evdata = readmatrix('EV_data.csv');

% Make solardata the same length as loaddata
solar_data = data.solarpower_15;
solar_data(numel(loaddata)) = 0;

% Delete last element of evdata to make same length
evdata(35041) = [];

% Get total load data
total_load = loaddata + evdata;

% Create x-axis for plot
x = [1:1:35040];

% Calculate excess energy generated by PV sytem
excess = solar_data - total_load;

% Create empty vector
Total_excess_day(1:365) = 0;

for z=1:365             % for every day of the year
    for y=1:96          % add every 15 minutes of a day
        if z>1
            % if not first day of the year, add the values incremented by
            % multiples of a day
            Total_excess_day(z) = Total_excess_day(z) + excess(y+(z-1)*96)*0.25;
        else
```

```
            % If first day of the year
            Total_excess_day(z) = Total_excess_day(z) + excess(y)*0.25;
        end
    end
end

% Create empty vector
Total_excess_jan = 0;
Total_excess_apr = 0;
Total_excess_jul = 0;
Total_excess_sep = 0;

for i = 0:1:6       % for 7 days, add all excess
    Total_excess_jan = Total_excess_jan + Total_excess_day(1+i);    % Day 1 is 1 jan
    Total_excess_apr = Total_excess_apr + Total_excess_day(104+i);  % Day 104 is 14 apr
    Total_excess_jul = Total_excess_jul + Total_excess_day(182+i);  % Day 182 is 1 jul
    Total_excess_sep = Total_excess_sep + Total_excess_day(254-i);  % Day 249 is 11 sep
end

% Output value
Total_excess_jan/7
Total_excess_apr/7
Total_excess_jul/7
Total_excess_sep/7

% Create figure
figure(1)
x = [0 365 365 0];
y = [0 0 1000 1000];
xlim([0 365])
f = figure;
f.Position = [10 10 1000 400];
plot(Total_excess_day)
xlabel('Time [days]')
ylabel('Energy shortage/excess [kWh]')
title('Average energy shortage/excess per day')
yline(0, '--black')
xline(1,'--black','1st of January');
xline(104,'--black','14th of April');
xline(182,'--black','1st of July');
xline(254,'--black','11th of September');
grid minor;
```

Listing A.1: Matlab code for calculating PV generation and BESS sizing

A.2. The Optimisation Model

```
from pyomo.environ import *
from pyomo.opt import SolverStatus, TerminationCondition
import pandas, numpy
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import seaborn as sns
import numpy as np

def readInputFile(filename):

    # Load all data from excel sheets

    LoadData = pandas.read_excel(filename, sheet_name= 'Load', index_col=0) #data of load entire neighbourhood
    TransformerData = pandas.read_excel(filename, sheet_name= 'Transformer', index_col=0) # rated power transformer
    PVProduction = pandas.read_excel(filename, sheet_name='PVProduction', index_col=0) # omgerekende irradiation
    StorageData = pandas.read_excel(filename, sheet_name= 'StorageSystem', index_col=0) # batterij
    EVDemand = pandas.read_excel(filename, sheet_name = 'EVDemand', index_col=0) #EV charging data

    # Return directory
```

```
    return {'PVProduction':PVProduction, 'StorageData':StorageData,
            'LoadData':LoadData, 'TransformerData':TransformerData, 'EVDemand':EVDemand}

def optimizationModel(inputData, modelType):

    # Unpack the data from the dictionary
    LoadData = inputData['LoadData']
    TransformerData = inputData['TransformerData']
    PVProduction = inputData['PVProduction']
    StorageData = inputData['StorageData']
    EVDemand = inputData['EVDemand']
#-------------------------------------------------------------------------------
    # Define the Model
    model = ConcreteModel()

#-------------------------------------------------------------------------------
    #Define Sets
    model.T = Set(ordered=True, initialize=LoadData.index)  # Set for time
    model.B = Set(ordered=True, initialize=StorageData.index)  # Set for battery
    model.G = Set(ordered=True, initialize=TransformerData.index) # Set for (grid) transformer

#-------------------------------------------------------------------------------
    #Define Parameters
        # Battery Energy Storage System
    model.BESS_Pmax = Param(model.B, within=NonNegativeReals, mutable=True)
    model.BESS_SOEmax = Param(model.B, within=NonNegativeReals, mutable=True)
    model.BESS_SOEini = Param(model.B, within=NonNegativeReals, mutable=True)
    model.BESS_Eff = Param(model.B, within=NonNegativeReals, mutable=True)
        # Load
    model.Consumption = Param(model.T, within=NonNegativeReals, mutable=True)  # Consumption of load j
        # PV Generation
    model.PV = Param(model.T, within=NonNegativeReals, mutable=True)  # Production of PV system k
        # Transformer
    model.Pmax = Param(model.G, within=NonNegativeReals, mutable=True)
        # EV
    model.EV = Param(model.T, within=NonNegativeReals, mutable=True)

#-------------------------------------------------------------------------------
    # Initialize Parameters
        # BESS Parameters
    for b in model.B:
        model.BESS_Pmax[b] = StorageData.loc[b,'Pmax']
        model.BESS_SOEmax[b] = StorageData.loc[b,'SOEmax']
        model.BESS_SOEini[b] = StorageData.loc[b,'SOEini']
        model.BESS_Eff[b] = StorageData.loc[b,'Eff']

        # PV generation Parameter
    for t in model.T:
        model.PV[t] = PVProduction.loc[t,'PVProduction']

        # Transformer Parameter
    for g in model.G:
        model.Pmax[g] = TransformerData.loc[g,'Pmax']

       # The timestep of the model
    timestep = 0.25

        # Load Parameter
    for t in model.T:
        model.Consumption[t] = LoadData.loc[t,'LoadData']

        # EV Parameter
    for t in model.T:
        model.EV[t] = EVDemand.loc[t, 'EVDemand']

#-------------------------------------------------------------------------------
    # Define the Decision Variables
        # BESS
```

```
    model.SOE = Var(model.B, model.T, within=NonNegativeReals)  # SOE
    model.Pch = Var(model.B, model.T, within=NonNegativeReals)  # P charge
    model.Pdis = Var(model.B, model.T, within=NonNegativeReals) # P discharge
    model.u_bess = Var(model.B, model.T, within=Binary)         # 1 is charging
    model.u_idle = Var(model.B, model.T, within=Binary)         # 1 is idleing
        # Transformer
    model.Pgrid_plus = Var(model.T, within=NonNegativeReals)    # pushing from
    model.Pgrid_minus = Var(model.T, within=NonNegativeReals)   # pulling to
    model.u_grid = Var(model.T, within=Binary)                  # 1 is pulling

    # Added in order to have curtailment
    model.curtail_pv = Var(model.T, within=NonNegativeReals)    # Curtail % of PV
    model.PVprod = Var(model.T, within=NonNegativeReals)        # PV Production

        # Added for EV curtailment/different wattages
    model.curtail_ev = Var(model.T, within=NonNegativeReals)    # Curtail % of EV
    model.EVDelayed = Var(model.T, within=NonNegativeReals)
    model.EVTotal = Var(model.T, within=NonNegativeReals)
    model.EVSup = Var(model.T, within=NonNegativeReals)
    model.SOPSup = Var(model.T, within=NonNegativeReals)
    model.SOPDemand = Var(model.T, within=NonNegativeReals)

#-------------------------------------------------------------------------------
    # Define Constraints

    def CurtailEV(model, t):
        if model.T.ord(t) == len(model.T):
            return model.curtail_ev[t] == 1
        else:
            return model.curtail_ev[t] <= 1

    def TotalEVDemand(model, t):
        if model.T.ord(t) == 1:
            return model.EVTotal[t] == model.EV[t]
        if model.T.ord(t) > 1:
            return model.EVTotal[t] == model.EV[t] + model.EVDelayed[model.T.prev(t)]

    def SupplyEVNow(model, t):
        return model.EVSup[t] == model.EVTotal[t] * model.curtail_ev[t]

    def DelayEV(model, t):
        return model.EVDelayed[t] == model.EVTotal[t] - model.EVSup[t]

    def TotalSupply(model, t):
        if model.T.ord(t) == 1:
            return model.SOPSup[t] == model.EVSup[t]
        if model.T.ord(t) > 1:
            return model.SOPSup[t] == model.SOPSup[model.T.prev(t)] + model.EVSup[t]

    def TotalDemand(model, t):
        if model.T.ord(t) == 1:
            return model.SOPDemand[t] == model.EV[t]
        if model.T.ord(t) > 1:
            return model.SOPDemand[t] == model.SOPDemand[model.T.prev(t)] + model.EV[t]

    def ForceCharge(model, t):
        if model.T.ord(t) > 8:
            return model.SOPDemand[t-8] <= model.SOPSup[t]
        else:
            return Constraint.Skip

    def ObjectiveFcn(model):
        return timestep*sum(model.Pgrid_plus[t] + model.Pgrid_minus[t] for t in model.T)

    # Added in order to have curtailment
    def CurtailPV(model, t):
        return model.curtail_pv[t] <= 1

    # Curtail > 0 not needed since its NonNegativeReal

    def PVcurtail(model, t):
```

```
        return model.PVprod[t] == model.PV[t] * model.curtail_pv[t]

    def PGrid(model, b, t):
        return model.Pgrid_plus[t] - model.Pgrid_minus[t] == model.Consumption[t] + model.Pch[b,t] \
                   + model.EVSup[t] - model.PVprod[t] - model.Pdis[b, t]

    def GridPull(model, g, t):
        return model.Pgrid_minus[t] <= model.Pmax[g] * model.u_grid[t]

    def GridPush(model, g, t):
        return model.Pgrid_plus[t] <= model.Pmax[g] * (1-model.u_grid[t])


    def SOE(model, b, t):
        if model.T.ord(t) == 1:
            return model.SOE[b,t] == model.BESS_SOEini[b] + timestep * (model.Pch[b,t] \
                                    * model.BESS_Eff[b] - model.Pdis[b,t]/model.BESS_Eff[b])
        if model.T.ord(t) > 1:
            return model.SOE[b,t] == model.SOE[b, model.T.prev(t)] + timestep * (model.Pch[b,t] \
                                    * model.BESS_Eff[b] - model.Pdis[b,t]/model.BESS_Eff[b])

    def BESS_SOE_max(model, b, t):
        return model.SOE[b,t] <= model.BESS_SOEmax[b]

    # BESS_SOE_min is not needed, since model.SOE specifies NonNegativeReals

    def BESS_Charging(model, b, t):
        return model.Pch[b,t] <=  model.BESS_Pmax[b] * model.u_bess[b, t]

    def BESS_Discharging(model, b, t):
       return model.Pdis[b, t] <= model.BESS_Pmax[b] * (1-model.u_bess[b, t])

    def BESS_idle(model, b, t):
        return model.u_bess[b,t] + model.u_idle[b,t] <= 1


#------------------------------------------------------------------------------
    # Add Constraints to the model

    model.Obj = Objective(rule=ObjectiveFcn)
    model.ConPGrid = Constraint(model.B, model.T, rule=PGrid)
    model.ConGridPull = Constraint(model.G, model.T, rule=GridPull)
    model.ConGridPush = Constraint(model.G, model.T, rule=GridPush)
    model.ConSOE = Constraint(model.B, model.T, rule=SOE)
    model.ConSOEmax = Constraint(model.B, model.T, rule=BESS_SOE_max)
    model.ConBESSCharging = Constraint(model.B, model.T, rule=BESS_Charging)
    model.ConBESSDischarging = Constraint(model.B, model.T, rule=BESS_Discharging)
    model.ConBESSidle = Constraint(model.B, model.T, rule=BESS_idle)

    # Added in order to have curtailment
    model.ConCurtailPV = Constraint(model.T, rule=CurtailPV)
    model.ConPVcurtail = Constraint(model.T, rule=PVcurtail)

    # Added in order to have curtailment of EV
    model.ConCurtailEV = Constraint(model.T, rule=CurtailEV)
    model.ConTotalEVDemand = Constraint(model.T, rule=TotalEVDemand)
    model.ConSupplyEVNow = Constraint(model.T, rule=SupplyEVNow)
    model.ConDelayEV = Constraint(model.T, rule=DelayEV)
    model.ConForceCharge = Constraint(model.T, rule=ForceCharge)
    model.ConTotalSupply = Constraint(model.T, rule=TotalSupply)
    model.ConTotalDemand = Constraint(model.T, rule=TotalDemand)

    return model
```

Listing A.2: The optimisation model in Python using the Pyomo packages