BACHELOR THESIS
COMPUTING SCIENCE

RADBOUD UNIVERSITY

Using optimized electric vehicle charging for demand side management

Author:
L. J. van Rhijn
s4739736

First supervisor/assessor:
Erik Poll
erikpoll@cs.ru.nl

Second assessor:
Pol Van Aubel
pol.vanaubel@cs.ru.nl

January 17, 2020

Abstract

The automotive sector is changing. More and more people are buying electric vehicles. Charging all these vehicles will put a huge load on the electricity grid, increasing the peak demand on the electricity grid. The peak demand of today is far below what our electricity grid can handle, but when a significant amount of people would start to switch over to electric vehicles, this peak will increase and the electricity grid could become overloaded.

A possible solution for preventing such an overload could be a smart charging algorithm. Usually, electric vehicles charge as fast as possible, using their available capacity completely. A smart charging algorithm changes how electric vehicles charge, spreading out their charging needs over the time they are parked.

Another way of preventing overload and lowering the demand of electric vehicle charging is by charging with electricity that is generated nearby. Electricity generated in residential neighborhoods (usually by solar panels) is often not entirely used by that neighborhood, causing a surplus of electricity. This surplus of electricity can be used to charge electric vehicles, which also lowers the demand on the electricity grid.

This research will present a valley-filling method of managing the charging of electric vehicles such that the peak load on the electricity grid is lowered and as much locally produced green energy is used as possible. Lowering the already existing peak load created by normal residential activity is not in the scope of this research. We will manage electric vehicle charging on a per neighborhood basis, dividing the total electricity needs of all vehicles over the period they are parked. The presented method will be evaluated on a couple of goals, two already mentioned above. We will look at the decrease in peak load, usage of locally produced green energy, and charging times of the electric vehicles. We then show that our method of smart charging performs well based on these goals.

Contents

Chapter 1

Introduction

Our electricity grid is evolving. From using grey sources of energy, such as coal and gas, we are now moving towards a more green future using energy from the sun and wind. We are also changing the way we drive, adopting more Electric Vehicles (EVs) [5] every year. These EVs have a high impact on our current electricity grid, using a lot of energy in short time frames [22]. An example is the Tesla Model S that can be equipped with a 20 kW charger. To put that in perspective, it uses much more energy than a regular television, which only uses around 100 Watts (0,1kW). Our electricity grid is not designed to deal with the extra load caused by EV charging, which is why we need to come up with a solution.

One solution might be to expand the current electricity grid so that it is capable of dealing with higher peak demand. An expansion of the current electricity grid is, however, very costly. Smarter ways to lower the impact of EVs on the grid are, for example, by using smart charging algorithms that spread out the total charging needs of EVs over the period they are parked. Another way to lower the extra load on the electricity grid is by using locally produced electricity for charging. The electricity produced in a neighborhood should be used within the same neighborhood. Using locally produced electricity is favorable because the generated electricity does not have to travel as far over the electricity grid as it would if it were not used locally, reducing the impact on the electricity grid.

Wijkie, a start-up company, is trying to achieve these positive effects of local power usage by sending people notifications when they should (and should not) use electricity. They have established a system where energy usage and generation in a neighborhood are monitored and processed by an algorithm. The algorithm sends out push notifications based on the current power level within the neighborhood, activating people to use more electricity when excess electricity is available. Wijkie wants to expand its algorithm by controlling smart EV chargers to regulate power usage, letting cars charge faster if a lot of energy is available. Electric vehicles could then be used to store this locally generated green electricity before notifying residents of the neighborhood that they should start using more electricity.

In chapter 2, we will look at Wijkie's current system and provide some background information on EV charging. In chapter 3, we will propose an algorithm to deal with smart EV charging and introduce goals that we will use in later chapters to evaluate the algorithm. In chapter 4, we will elaborate upon our testing environment and show the used test data. Chapter 5 will go into the evaluation of the goals established in chapter 3 and will provide an overall evaluation of the algorithm. Chapter 6 will provide some background reading material about smart EV charging and chapter 7 will provide some future research suggestions. A conclusion will be given in chapter 8.

Chapter 2

Background

This chapter contains background information about the topics discussed in this paper. Wijkie and its current algorithm will be elaborated upon in section 2.1. Section 2.2 will give an overview of the current EV charging landscape and will show how Wijkie will fit into this landscape. Section 2.3 will go into the different methods of charging EVs and will show the method we will be using in the next chapters.

2.1 Wijkie

Wijkie (previously known as Groenewijkstroom) is an organization that deals with consumer-generated green energy. More specifically, if a lot of green energy is produced in a neighborhood, this energy should be used within the same neighborhood. Wijkie offers an application to its customers that sends out a push notification if more energy is produced than is used within the customer's neighborhood. The push notification leads to residents using more energy when green energy is available and the opposite. Electricity companies do not have to compensate as much as people change their electricity usage behavior depending on the level of available green energy. Another benefit lies in the fact that our current electricity grid is not built for the amount of load that EVs and green energy sources will put on the grid. Consuming locally generated electricity is a good way of reducing the load on the electricity grid.

As previously said, Wijkie has an application that can be installed on mobile phones. The application notifies people of energy surplus in a neighborhood through push notifications. It all works by placing a self-designed device in smart electricity meters in residential neighborhoods. This device sends data to a central server which then computes the electricity usage in a specific area. More on this will be explained in section 2.1.1.

Customers of Wijkie can earn "Wijkies", a virtual coin that can be spent within a customer's neighborhood. These Wijkies are earned every time a user decides to use electricity when there is a surplus in the neighborhood. Wijkie partners up with local entrepreneurs to grow the number of stores that Wijkies can be spent at.

2.1.1 Wijkie system overview

Currently, Wijkie uses a self-designed device (further called EARN-E) to pull electricity usage and generation data from smart electricity meters. The EARN-E connects to a WiFi hotspot within the house, which connects to a central database that stores all electricity usage data from Wijkie's customers. The usage data is pulled from the P1 port on smart electricity meters. The database connects to the Wijkie app and sends different kinds of data back and forth between the app and the server. The information flows are shown in figure 2.1.

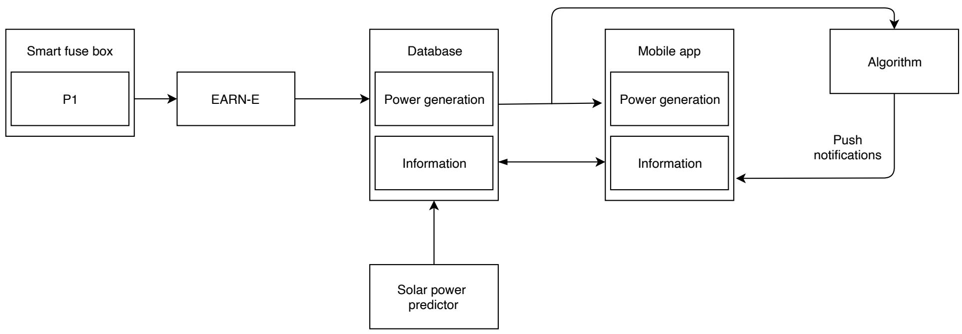

Figure 2.1: Current (trimmed down) diagram of Wijkie's system

As we can see, the algorithm only uses power usage and generation data to calculate whether or not to send push notifications. The solar power predictor is not used within the algorithm. Its sole purpose is to graphically show the expected amount of photovoltaic energy that will be generated within the app.

The algorithm used for sending push notifications to Wijkie's app users is relatively simple. When electricity generation within a neighborhood is higher than the electricity used within the same neighborhood, all people in the neighborhood receive a push notification. The same thing happens whenever the generated electricity drops to half that of the electricity used within the neighborhood. We established a formula for the algorithm in figure 2.2.

$$P_n^t = \begin{cases} S_1 & \text{if } G_n^t > U_n^t \wedge ((\forall_{s<t} P_n^s \neq S_1 \wedge P_n^s \neq S_2) \vee \\ & (\exists_{s<t} P_n^s = S_2 \wedge \forall_{s<r<t} P_n^r \neq S_1)) \\ S_2 & \text{if } U_n^t > 2 * G_n^t \wedge \exists_{s<t} P_n^s = S_1 \wedge \forall_{s<r<t} P_n^r \neq S_2 \\ N & \text{otherwise} \end{cases}$$

Figure 2.2: Formula used by Wijkie's algorithm

The terms used in the formula are explained below.

P_n^t = Push notification in neighborhood n at time t
S_1 = Signal 1, a notification recommending users to use more energy
S_2 = Signal 2, a notification recommending users to use less energy
N = No notification
G_n^t = The energy that is generated within neighborhood n at time t
U_n^t = The energy that is used within neighborhood n at time t

A graphical example of an arbitrary neighborhood a is displayed in figure 2.3. The green line represents G_a^t and the black line represents U_n^t. On time $t = 3$, signal one will be dispatched to the users because $G_a^3 > U_a^3$ and for all $t < 1$, $P_n^t \neq S_1$ and $P_n^t \neq S_2$. The second case is true for time $t = 15$ as $U_a^{15} > 2 * G_a^{15}$ and there exists a t for which $P_a^t = S_1$ (namely $t = 3$) and for all times t in between $t = 3$ and $t = 15$ $P_n^t \neq S_2$.

2.2 EV charging landscape

The commercial EV charging landscape is complex, with many parties regulating different parts and many protocols supporting communication between these parties. We will give a brief overview of all the parties involved with commercial charging poles. This overview will rely heavily on research by Poll and Van Aubel [26]. For a complete overview, we advise reading the original paper.

The central parties involved with EV charging are:

- Charge point operators (CPO) operate and maintain charge points.
- E-Mobility service providers (eMSP) resell electricity to consumers.
- Distribution system operators (DSO) manage the electricity grids.

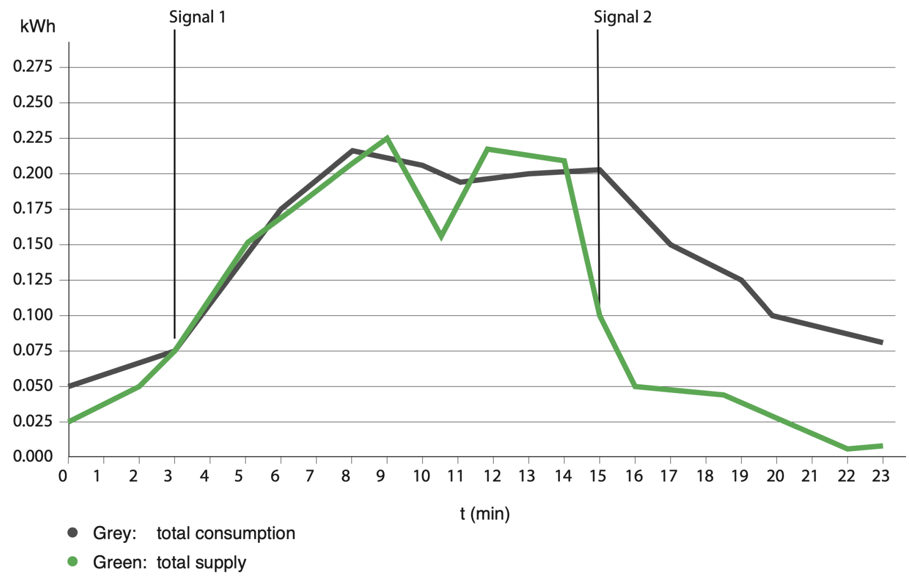

Figure 2.3: Example of Wijkie's algorithm

- Clearing houses (CH) offer a standardized platform to communicate between CPOs and eMSPs (**A** in figure 2.4). This communication can also take place without a CH (**B** in figure 2.4).
- Electricity suppliers (ES) provide the electricity needed to charge EVs.
- Charge point infrastructure operators (CPIO) produce and maintain charge points. The CPO can also conduct some of the tasks of a CPIO.

A graphical representation of the EV charging landscape can be found in figure 2.4.

There are many parties involved in commercial EV charging, making the landscape rather complex. To make it even more complex, all these parties have different protocols to communicate with each other. Important protocols to note are Open Charge Point Protocol (OCPP) [19] and Open Smart Charging Protocol (OSCP) [20]. OCPP allows CPOs to communicate with the charge points (**C** in figure 2.4). The protocol offers a standardized way to communicate with supported charge points. OCPP is used to control remote maintenance of charge points, but it can also be used to set a maximum load per time frame for charge points. With OCPP, we can thus control how much current a charger pulls from the electricity grid.

OSCP is used to provide a prediction of the electricity load on a per cable basis. OSCP provides predictions based on historical electricity usage data.

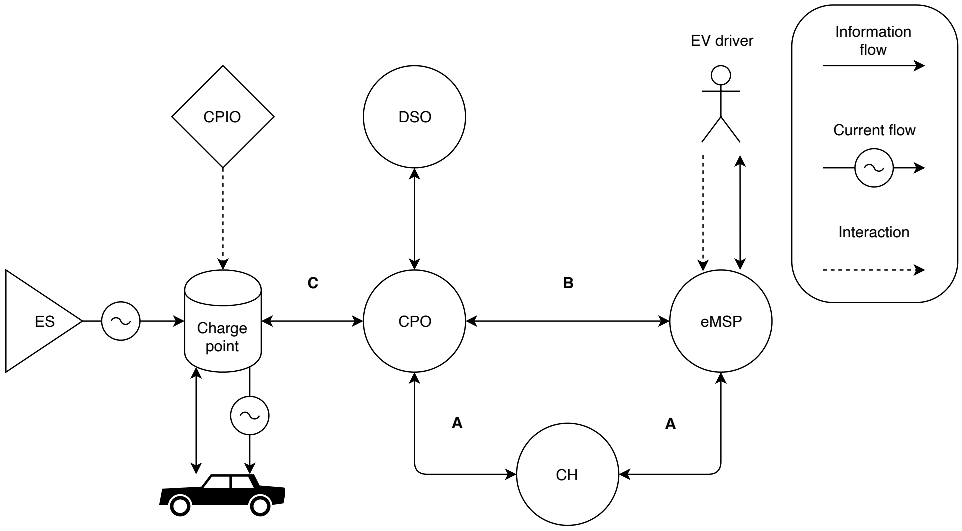

Figure 2.4: EV landscape graphically

Historical data is useful for charging EVs because we want to be able to predict when there is capacity available to charge connected EVs. More on this will be explained in chapter 3.4.1.

The role of Wijkie in this ecosystem can be resembled by a CPO and an eMSP, which can be the same entity, according to Poll and Van Aubel [26]. Wijkie will be in control of the charging points by sending charge plans to the charging points. They will also interact with their customers to provide them with their services.

2.3 Different ways of smart charging

Smart charging is the intelligent charging of EVs. Charging can be shifted based on grid loads and in accordance with the vehicle owner's needs. There are many ways in which EV charging can occur and which party controls what EV to charge. This section will explain some of the different ways smart charging can be regulated. We will summarize which types of smart charging will be used in the remainder of this paper in section 2.3.4.

2.3.1 Uni- and bidirectional charging

At first, we can distinguish between unidirectional and bidirectional (also called vehicle to grid, V2G) charging. Unidirectional charging means normal charging, as one would charge a smartphone. Just plug it into a power outlet and begin charging until the battery is full. Bidirectional charging allows for the EV to supply electricity back to the power grid, meaning we could regulate the power flow within the power grid by smartly (dis)charging EVs. Bidirectional charging will not be further discussed as the implementation of bidirectional charging is not relevant for Wijkie. Bidirectional charging is still in development, and there is also concern that bidirectional charging has much impact on the EV's battery life [6].

2.3.2 Peak shaving and valley-filling

The terms peak shaving and valley-filling are often used with different meanings in different papers [8][12]. We will use the two terms in the following way. Peak shaving is used when talking about lowering the normal, already present, load peaks on the grid by using connected EVs. An existing peak will be lowered by discharging EVs when demand is high. This way, EVs act like a sort of battery attached to the power grid. For peak shaving to work, an EV needs to be capable of supplying power back to the electricity grid. As already noted in section 2.3.1, bidirectional charging is not in the scope of this paper as the impact of bidirectional charging on the battery lifespan of an EV is still being researched [6].

Valley-filling is another method of regulating the electricity load so that the grid does not become overloaded. Valley-filling describes methods to lower the extra load created by charging EVs and spreading out that load over some time. Valley-filling algorithms do not necessarily use vehicle-to-grid technologies but only control the amount of energy an EV charger can use from the electricity grid at specific time intervals. This research will include a method of valley-filling to decrease the extra load EVs put on the electricity grid.

2.3.3 Centralised and decentralized algorithms

Another distinction can be made in ways to decide which vehicles to charge and the information available. A global distinction can be made between centralized and decentralized methods of smart charging. A centralized way usually means that a central party or server controls all EVs within an area. A benefit of these central algorithms is that an optimized charging schedule can be found such that the electricity load is lowest. Central algorithms usually also have a complete set of data available to them and not only a small set found in local or decentralized algorithms. A downside of central algorithms lies in the fact that computational time is usually much longer than that of a decentralized algorithm; this is also the most used argument to create decentralized approaches to smart charging.

A decentralized (or local) algorithm usually consists of a lot of different agents that run the same algorithm. These different agents either only use their local data to compute an optimal charging plan [17][30] or communicate with each other to come up with the most optimal charging plan for a larger area [27]. The benefits of decentralized approaches are scalability and, in general, much faster computational times. A downside is that often much infrastructure must be placed in order for these algorithms to work.

To achieve a kind of system that uses a local algorithm would mean that either the EARN-E has to be expanded to an extent where it can communicate with the EV charger, or another device has to be put inside the homes of Wijkie's customers. As we will also see later on, a centralized approach (based on already existing infrastructure Wijkie has) is more desirable than a decentralized approach. The algorithm proposed in this paper will thus be centralized.

2.3.4 Approach used in this paper

Summarizing the points made in previous sections, our algorithm will be a *centralised unidirectional* smart charging algorithm that accomplishes *valley-filling*. The algorithm will compute charging plans on a central server run by Wijkie. This central algorithm collects all data send by the EARN-E devices and will come up with the most optimal charging plan for the next hours. A complete specification of the algorithm can be found in chapter 3.

Chapter 3

Demand side EV management

This chapter will start with a quick explanation of the proposed algorithm in section 3.1. Section 3.2 will explain the problems noted in previous chapters in more detail, specifically explaining valley-filling. It will also state the goals that the algorithm needs to accomplish. In section 3.3, we will discuss the formalisation used in the following chapters. Section 3.4 will discuss some of the design decisions made to implement the algorithm into the current infrastructure. Section 3.5 will explain the proposed algorithm in detail.

3.1 Algorithm summary

This section will provide a brief overview of the algorithm proposed in section 3.5. When electric vehicles are connected to the power grid, they will send a message to a central server that will record what EVs are connected in which neighborhoods. The proposed algorithm will be executed on a per neighborhood basis, only the connected EVs in a neighborhood are considered. The EVs will pass on the amount of electricity they need and when they expect to leave their chargers. The algorithm then splits the amount of needed power in chunks of a standard size. These chunks will then be allocated on time frames where the load is lowest, also keeping in mind when the EVs need to leave. This way, the total load on the electricity grid is divided over the amount of time that the EVs are expected to be charged instead of charging at full capacity when EVs are connected.

3.2 Goals

There are four goals we want to achieve with the new smart charging system, listed in order of importance:

1. The EVs should be charged fully when the (expected) departure time of the EV approaches.
2. The system should decrease the peak load on the electricity grid created by simultaneously charging multiple EVs.
3. The EVs should be charged relatively fast until hitting 80% battery capacity[1]. This type of fast charging is needed because there may be times when people suddenly need their vehicle.
4. The system should prefer using green energy generated nearby when available.

The first goal is that the EVs must be fully charged when their expected departure times approach. Without the first goal, it would be rather trivial to create an algorithm that decreases the grid load created by EV charging as we could just not charge the EVs at all. This first goal only applies if it is physically possible to charge the EV before its departure time (keeping in mind the maximum load of the charger and the capacity the EV has to charge).

The second goal we want to accomplish by regulating EV chargers is valley-filling. We want to lower the peak load the electricity grid needs to handle by spreading out that load over a larger time frame (explained in section 2.3.2). In order to decrease the peak load on the electricity grid, we need to shift EV charging to times where electricity demand is low. As an example, figure 3.1 displays the average electricity usage of a household on a typical day [28].

The peak load time is around 18:00. People start coming home from their work and turning on all different kinds of electrical devices. The maximum load the electricity grid is capable of handling is higher than the peak load occurring at around 18:00 each day. A problem occurs when the peak load exceeds the maximum load the electricity grid is capable of handling. When electricity demand exceeds the maximum load, power outages can occur. We should thus avoid creating a higher peak load as this might result in an overloaded electricity grid.

The third goal we want to accomplish has to do with the usability of the smart charging algorithm. As we can see in section 3.3, we need a point in time where the EV needs to be fully charged. This point in time might

[1] Based on iPhone smart charging [1].

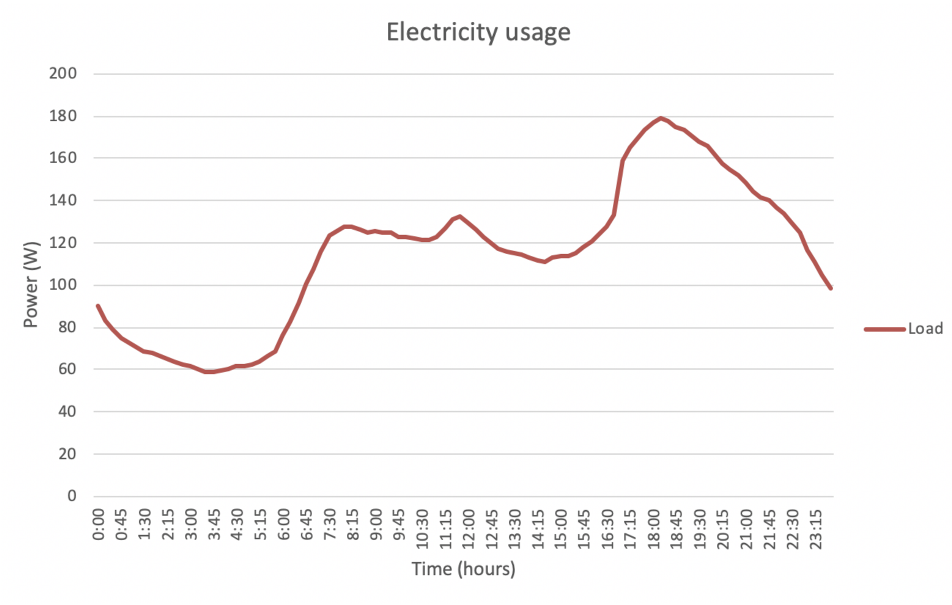

Figure 3.1: Average electricity usage of a household on a typical day

be predicted or entered by the user. The problem is that an EV might be charged right before the EV needs to be fully charged. When this is the case, and the user unexpectedly needs their EV earlier than its original leaving time, the EV might not be ready to depart because of a non-charged battery. The time it takes to fully or partially charge an EV is thus important for the usability of our algorithm. We will take a look at the charging times of the EVs and see how our smart charging algorithm performs.

The last goal we want to accomplish is using as much green energy within a neighborhood as possible to charge EVs. Using surplus green energy for charging EVs is beneficial in two ways. First off, Wijkie does not have to send as many push notifications via their app as our algorithm already uses the surplus green energy by charging the EVs. The second benefit is that the load on the electricity grid created by green energy generation is lowered.

There might be a problem with achieving the last goal. Green energy generated within a neighborhood is, most of the time, photo-voltaic energy (energy generated from sunlight). We predict that most of the EVs will be away from home when solar energy is produced as they are most likely being used as commuting vehicles, which would mean that there might not be an opportunity to charge these commuting EVs with solar energy. To charge cars with sunlight, they need to arrive at home before or during the generation of photo-voltaic energy. We will use non-commuting vehicles to see whether our algorithm is capable of dealing with energy generation within the neighborhood. These non-commuting vehicles will be further explained in section 4.3.

3.3 Formalisation

A possible solution for a valley-filling algorithm has been proposed by Vandael et al. [27]. The authors solve the issue by flattening the load on both high and low voltage transformers. The load flattening is done by sending reservation requests back and forth between these transformers and the EV charging units. Both the high and low voltage transformers then execute an algorithm to decrease peak load as much as possible. We will use an adaptive version of the algorithm to flatten the load in neighborhoods. The examples that will be used in the next sections will consider computations in an arbitrary neighborhood.

3.3.1 Terminology

The following terms will be used:

- U_H^t: The electricity balance at time t for all households in a neighborhood. Usage and generation of all households in a neighborhood are included. EV charging usage is excluded.
- C_e^{total}: Total capacity of EV e (for a fully charged battery).
- C_e^{now}: Current capacity of EV e.
- $C_e^{charge} = C_e^{total} - C_e^{now}$.
- t_e: Expected leave time for EV e.
- L_e^{max}: The maximum load of the charging pole connected to EV e. This variable depends on the maximum load the EV and the charging pole can draw from the electricity grid.
- L_e^t: The amount of electricity that the charging pole connected to EV e uses at time t.
- AU^{min}: The minimum allocation unit constant. This variable is constant and will be set before the algorithm runs. AU^{min} is further explained in section 3.5.
- A: The number of time frames to allocate for. This variable is constant and will be set before the algorithm runs.
- t_w: The time frame width. This variable is constant and will be set before the algorithm runs.

Two other terms that we will frequently use are allocation space and the prediction window. With allocation space, we mean the time we have in which we can allocate charging for an EV. It is the time between $t = 0$ and $t = t_e$.

With the prediction window, we mean the amount of time the algorithm maximally looks into the future for allocating EV charging. When the leaving time of an EV is further in the future than the prediction window, EV charging will be allocated as if its leaving time is the same as the end of the prediction window. The prediction window is equal to $A * t_w$.

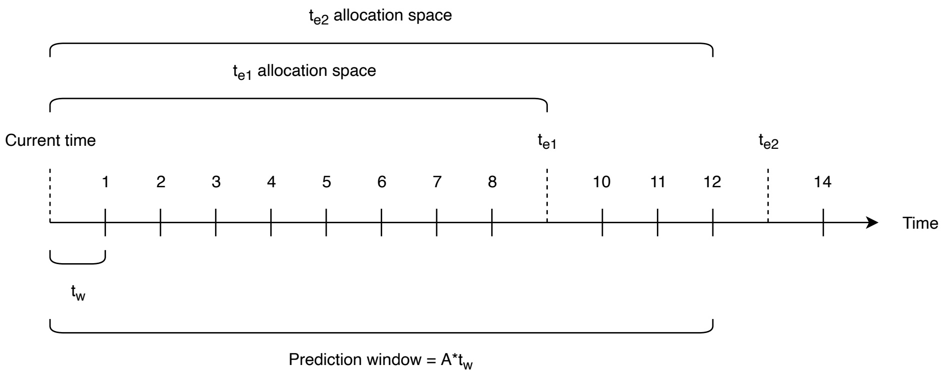

Figure 3.2: Prediction window example

Figure 3.2 displays an example of a prediction window with $A = 12$ and $t_w = 1$ time step. The leaving time of EV $e1$ is set at time step 9 and the leaving time of EV $e2$ is set at time step 13. The electricity car $e1$ needs for a fully charged battery will be allocated between time step 0 and 9 because time step 9 is in range of the prediction window. Car $e1$'s allocation space is thus between $t = 0$ and $t = 9$. Car $e2$'s leaving time is not in range of the prediction window. This means that the algorithm will allocate $e2$'s charging within range of the prediction window, as if $t_{e2} = 12$. Car $e2$'s allocation space is thus between $t = 0$ and $t = 12$.

3.3.2 Input

As input to our algorithm, we expect to receive the following:

- U_H^t for all t in the the prediction window. All future U_H^t will be predicted. This will be further explained in section 3.4.1.
- C_e^{total} for all EVs e in the neighborhood connected to a charging pole.
- C_e^{now} for all EVs e in the neighborhood connected to a charging pole.
- t_e for all EVs e in the neighborhood connected to a charging pole.

- L_e^{max} for all EVs e in the neighborhood connected to a charging pole.

3.3.3 Output

As the output of our algorithm, we expect to calculate the following values:

- L_e^t for all e in the neighborhood connected to a charging pole and all t in the prediction window.

3.4 Integration in the current infrastructure

There are a couple of things still left unclear about how our algorithm would integrate into the current infrastructure. The prediction of U_H^t will be explained in section 3.4.1, and the data flow of our algorithm will be explained in section 3.4.2.

3.4.1 Prediction of household electricity balance

As specified in section 3.3.2 and 3.4.2, we need to be able to predict the usage of houses within a neighborhood. We can do this in two ways:

1. We can use the OSCP protocol already mentioned in section 2.2.
2. We can use data stored in Wijkie's database to make our own predictions.

The first option relies on an implementation of the OSCP protocol within our algorithm. A DSO is able to predict the amount of capacity that is available on a per cable basis for specific time intervals and transmit that information to us via OSCP. An overview of this implementation is given in figure 3.3.

A significant disadvantage of using the prediction of a DSO for our algorithm is that the minimal time interval for these predictions is fifteen minutes.

The second option we have is to use data that Wijkie already has. Wijkie collects data from the P1 port on smart electricity meters and stores this data on their central server on a per-minute basis. With a large enough data set of P1 data, Wijkie could be able to predict the electricity usage per household. The implementation of this option would be similar to the previous one, but would only require the database of P1 data that Wijkie has stored. An overview of this implementation is given in figure 3.4.

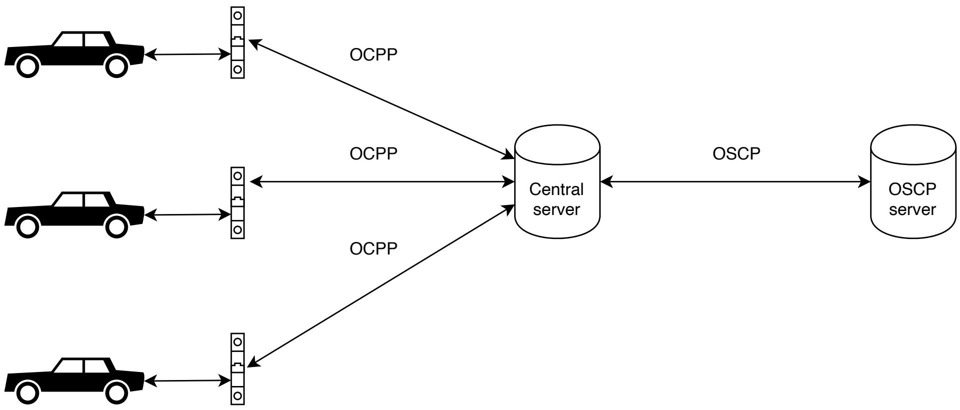

Figure 3.3: OSCP implemented within the proposed algorithm

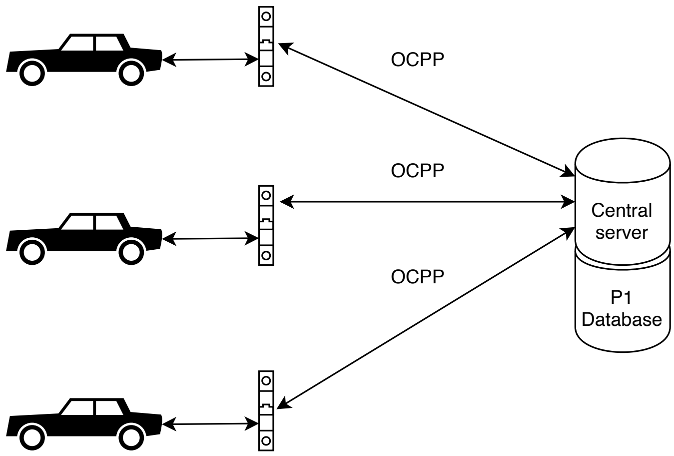

Figure 3.4: P1 Database implemented within the proposed algorithm

The prediction of electricity usage could be made with various methods, such as with neural networks [21], state-space models [11], or just the average over a couple of weeks.

The preferable way of predicting household electricity balance is by using the Wijkie database. By using the data Wijkie already has, we can create a more frequent (and maybe even more precise) prediction than a DSO can provide us over OSCP.

3.4.2 Data flow

The algorithm we will be using needs a lot of data from different parties. The car should, for example, provide the amount of power it needs to charge. A specification of the data flow for our algorithm is given in figure 3.5. An explanation will be given below.

1. The car is connected to the charger and will send its total battery capacity (C_e^{total}), its current capacity (C_e^{now}), and its maximum load (L_e^{max}) to the charge point.

2. The charge point collects the information sent by the car. It checks whether its maximum load is lower than that of the car. If its maximum load is lower than that of the car, it will send its own maximum load to the central server. Otherwise, it will forward the car's maximum load to the server. The charge point also adds an expected leaving time (t_e). The expected leaving time could be provided in other ways, which is discussed in section 7.3.

3. The central server runs the algorithm we will describe in chapter 3.5 at regular intervals. It thus waits a couple of minutes before the algorithm computation starts (**A** in figure 3.5). This time is used to collect information about all (new) charging cars. Cars are expected to send their status messages (**1** in figure 3.5) regularly. Regular status messages are necessary so that the algorithm knows which cars are connected and which cars left their charging stations. After a regular interval has passed, the central server will send the prediction window to the prediction server ($A * t_w$, A intervals of t_w time). The prediction server is expected to calculate the energy usage prediction for all time frames in the prediction window.

4. The prediction server returns a prediction for every time-frame t.

5. The central server runs the algorithm and tells all connected cars how much power they should draw (L_e^t) for the next time frame.

3.5 Proposed valley-filling algorithm

The algorithm we will use for valley-filling will allocate energy loads to specific time frames. These time frames can have any span but, for the following example, we will assume they are $t_w = 1$ hour each. At the start of each time frame, we will loop over all EVs that have requested to receive a charging plan and allocate charging accordingly by increasing loads that are smallest. The maximum load for every EV e is set at L_e^{max}. We will not allocate more energy to EV e within a time frame t than L_e^{max}, $L_e^t \leq L_e^{max}$. The algorithm does the following. For each car e:

1. Split up C_e^{charge} into pieces of minimum allocation unit. The AU^{min} is defined as a constant of, for example, 1kWh.

2. For each AU^{min}, do the following:

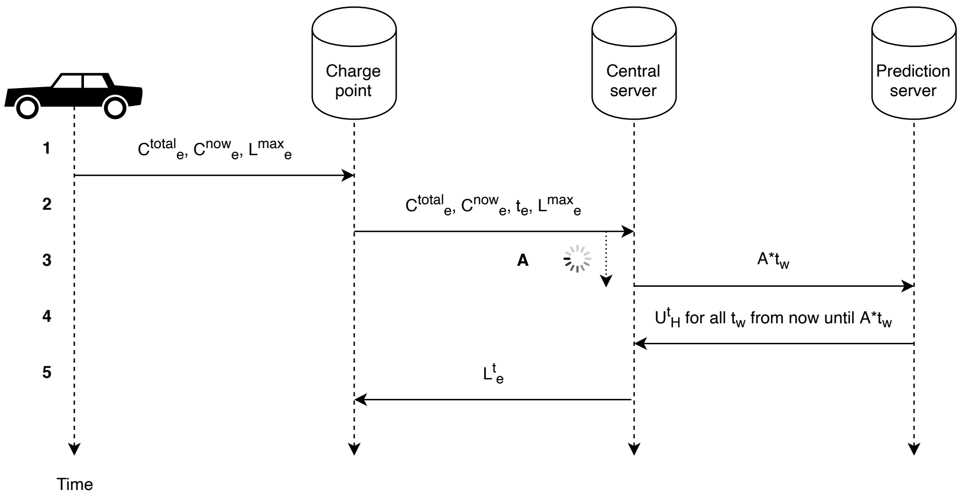

Figure 3.5: Data flow for our algorithm

(a) Search the time frame t_o with the lowest load on the network from $t = 1$ until $t = min(t_e, t_w * A)$ and with the already allocated load for EV e below L_e^{max}. If there is more than one time frame t_o, choose the earliest one.

(b) Reserve a minimum allocation unit on t_o.

Figure 3.6 displays an example.

As can be seen, the base household load is plotted. The algorithm will assign charging loads to time frames for every vehicle e. Let's say, for example, vehicle $e1$ needs $C_{e1}^{charge} = 4$kWh and has an expected leave time of $t_{e1} =$ 3 hours. The load of 4kWh is then split up and divided over the first and the second hour as these have the lowest loads in the allocation space of $e1$. Vehicle $e2$ also wants to charge, it needs $C_{e2}^{charge} = 8$kWh and has an expected leave time of $t_{e2} = 12$ hours. The maximum load of this vehicle is $L_{e2}^{max} = 2$kW (it can draw 2kWh per time frame). This means we have to divide the energy that needs to be charged into at least four different time frames because we can only charge 2kWh per time frame at maximum. We can see that the energy is split up into the time frames 4, 5, 6, 11, and 12.

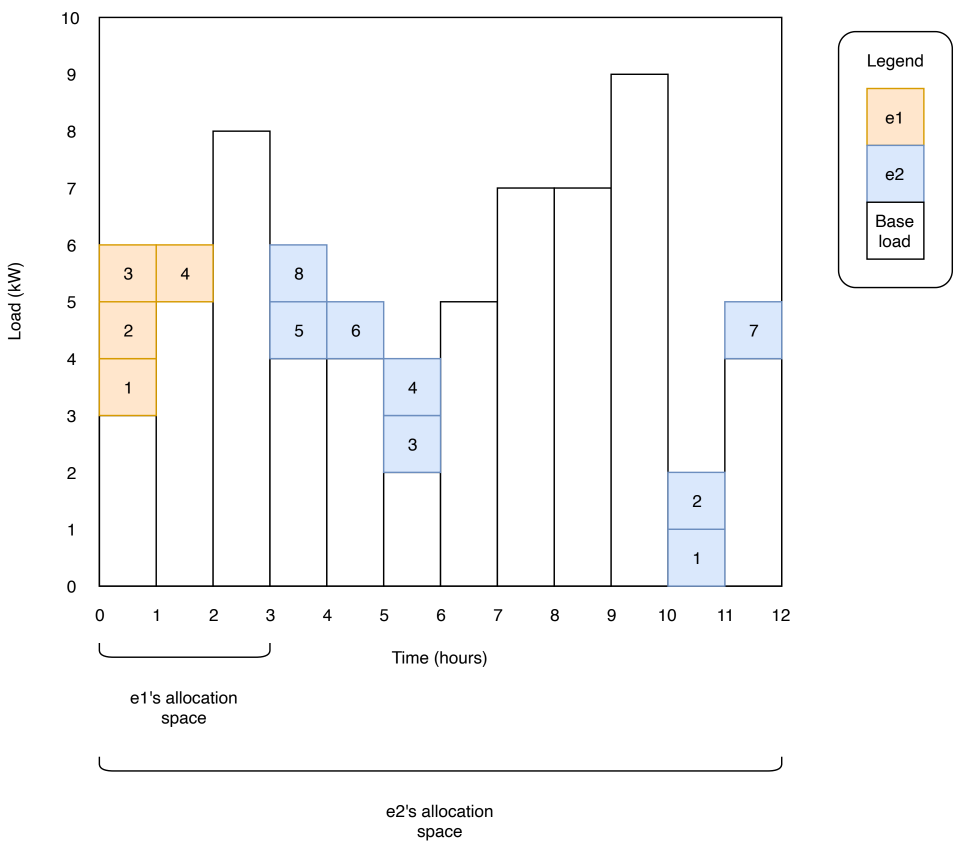

Figure 3.6: Algorithm example

Chapter 4

Simulation

This chapter will go in detail how the proposed algorithm will be designed and implemented into a simulation that is used in chapter 5 to test whether the goals in section 3.2 have been met. Section 4.1 will show the data used for the simulation. The prediction of electricity load data will be discussed in section 4.2. Specifications of the cars used in our simulation will be discussed in section 4.3, and the settings used for the simulation will be discussed in section 4.4.

4.1 Test data

The data that will be used for testing our algorithm in chapter 5 is electricity usage and generation data of twelve households with a time interval of five minutes. This household data will include several houses in Arnhem and Ede. We have aggregated the data and will use the usage minus generation in our simulation. A graphical example of the data can be found in figure 4.1.

As we can see, the average energy balance in figure 3.1 can be recognized in this diagram. There are peaks in usage around the morning and evening. Another thing that stands out is the valleys that reach negative values. These valleys are caused by solar panels generating electricity.

4.2 Simulating the prediction of electricity loads

Section 3.3.2 and 3.4.1 showed us that we need a prediction of the energy balance in the neighborhoods where our algorithm functions (U_H^t). We will be simulating these predictions when testing our algorithm. The predictor

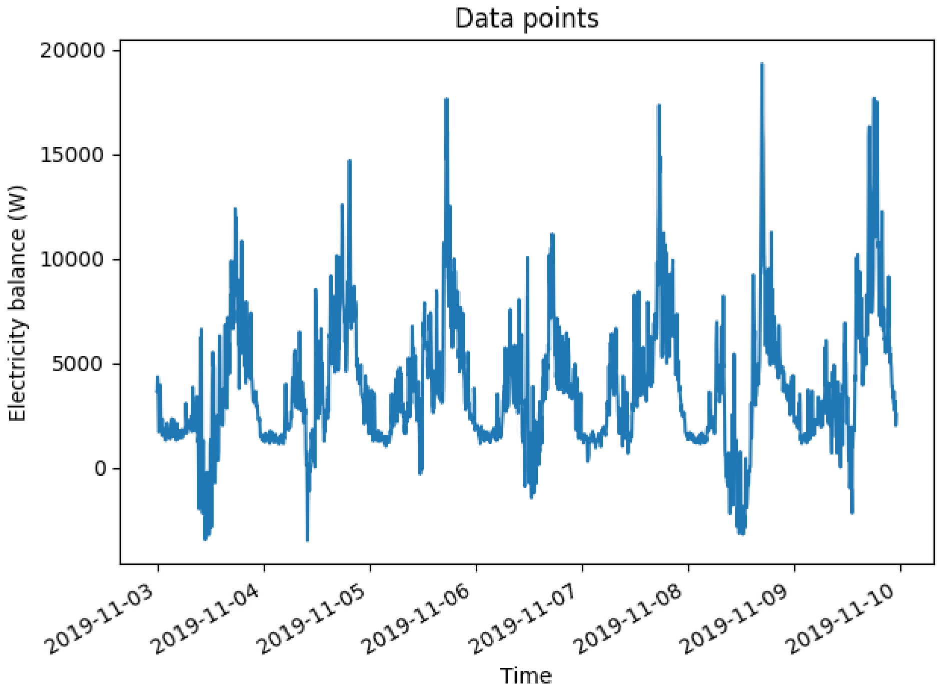

Figure 4.1: Test data

will do the prediction simulations. The predictor uses the test data discussed in section 4.1. The predictor creates predictions by taking the original test data and adding a random error, simulating real-life prediction performance.

Another feature built into the predictor of the simulation is one that allows us to specify a smaller error for time steps that are close by and a larger error for time steps further in the future. For example, if we make a prediction of twelve hours, we can specify the error rate of the first time step and the error rate of the last time step. The predictor will then linearly scale the error rate between the other time steps so that it creates a prediction with an ever-increasing error rate.

A prediction of twelve hours of the test data can be found in figure 4.2. The prediction data was generated by allowing a minimum error of 500W and a maximum error of 2kW.

We can see that the error, indicated by a black line, starts at 500W and increases until it reaches 2kW at the last data point. The prediction also starts close to the original test data, but increases in error rate for data points further in the future.

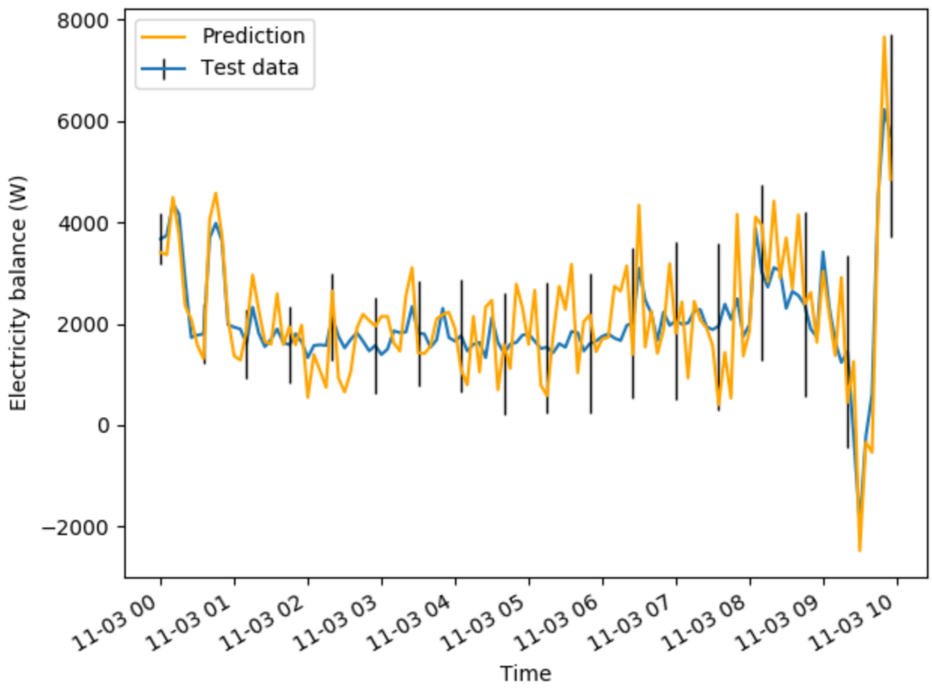

Figure 4.2: Prediction of 12 hours of electricity balance

4.3 Simulation of EVs

The simulation we will be running in chapter 5 will make use of a set of randomly generated cars. For each of these randomly generated cars, a cycle will be run each day. Each car will leave home and come back once every day. We make a difference between commuting (these leave early in the morning and come back in the afternoon) and non-commuting cars (these leave early in the afternoon and come back a couple of hours later). Commuting and non-commuting cars will have a different set of randomly generated values, simulating the different characteristics of both types of cars. A normal car cycle is graphically shown in figure 4.3.

The simulated cars will use a lot of randomly generated values to create a real-life scenario. These randomly generated values are used for computing the connection time, leaving time, and battery capacity left each trip. All randomly generated values will have an upper and lower bound, as can be seen in figure 4.3.

The generated cars all have the following properties:

1. The battery capacity of the car (C_e^{total}).

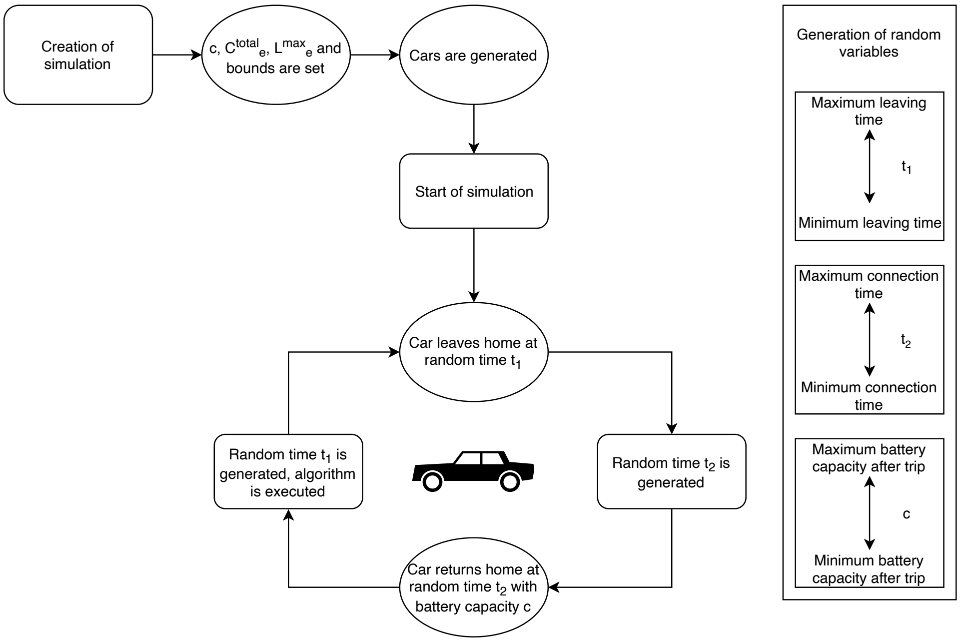

Figure 4.3: Cycle of a simulated car

The battery capacity of the car is randomly generated at the start of the simulation for each car. The upper and lower limits of the randomly generated battery capacity are set at the start of the simulation.

2. Leaving time of the car each day (t_e).

The leaving time of a car is generated when it arrives at home each day. The leaving time is a randomly generated value between a lower and upper bound. These bounds are set at the start of the simulation.

3. Connection time of the car each day.

As with the leaving time, the connection time of a car is randomly generated between a lower and upper bound. The connection time is generated when a car leaves home.

4. Maximum load the car can charge per time step (L_e^{max}).

The maximum load that the car can charge per time step is set at the start of the simulation.

5. Amount of battery the car drains each trip.

The amount of battery drained by the car each trip is generated at the start of the simulation and set after each cycle. The amount of battery left is a randomly generated value between a lower and upper bound. Both bounds are set at the start of the simulation.

4.4 Simulation settings

This section will describe the standard settings of the variables we will use when evaluating our algorithm in chapter 5. The variables are set in the following way:

- We will use electricity balance data from twelve households.
- We will generate four cars, three used for commuting and one used for non-commuting purposes.
- All commuting cars will leave at a random time each day. This random time has a lower bound of 06:00 and an upper bound of 09:00.
- All commuting cars will come home at a random time each day. This random time has a lower bound of 16:00 and an upper bound of 19:00.
- All non-commuting cars will leave at a random time each day. This random time has a lower bound of 11:00 and an upper bound of 12:00.
- All non-commuting cars will come home at a random time each day. This random time has a lower bound of 13:00 and an upper bound of 16:00.
- All cars have a fixed total capacity of 75kWh, based on the Tesla Model S specifications [7].
- All cars have a fixed maximum load. The maximum load is set at 11kW, based on the Tesla Model S specifications [7].
- All cars have a randomly generated capacity left each time they come home. This capacity is fixed throughout the simulation. All commuting cars have a randomly generated capacity between 50% and 90% (these numbers are based on a study about commuting [25]). All non-commuting cars have a randomly generated capacity between 70% and 90%.
- The simulation has a fixed prediction window. The prediction window is set at 12 hours.
- The simulation has a fixed minimum allocation unit. This minimum allocation unit is set at 250Wh.
- The simulation has a fixed error for the predictions made by the predictor (see section 4.2), this error is set at 500W for the first time step and 2kW for the last time step (based on the error rates in a paper by Sandels et al. [23]).

Some tests in chapter 5 will use other variable settings than the ones described above. Whenever this is the case, it is explicitly stated in the section that uses other variable settings.

Chapter 5

Evaluation of proposed algorithm

This chapter will go into detail about the performance of our proposed algorithm within the created simulation. We will first show some general test results and compare uncontrolled charging with controlled charging data in section 5.1. In section 5.2 we will evaluate the goals set in section 3.2 and see how our algorithm performs.

5.1 Test results

We will first show how the electricity grid balance varies over time when not using an algorithm to control EV charging. The EVs just start charging at their maximum when they are connected to the grid. We will compute results over six days of data. The results can be seen in figure 5.1.

We can see that the peak load on the electricity grid reaches 40kW easily on multiple days. The overall maximum is 45 576W. The maximum is as expected as the chargers of our cars are 11kW each and we have three cars charging at night. The extra load generated by the EVs more than doubles the electricity load on all days.

We will now run our algorithm to see its effect on the peak load. Figure 5.2 shows the result of one standard run while using our algorithm to optimize charging and figure 5.3 shows a comparison between optimized and non-optimized charging from figure 5.1 and figure 5.2.

We can see that the peak load on the electricity grid does not exceed the 19 308W maximum our test data also had. Electricity drawn from the grid is

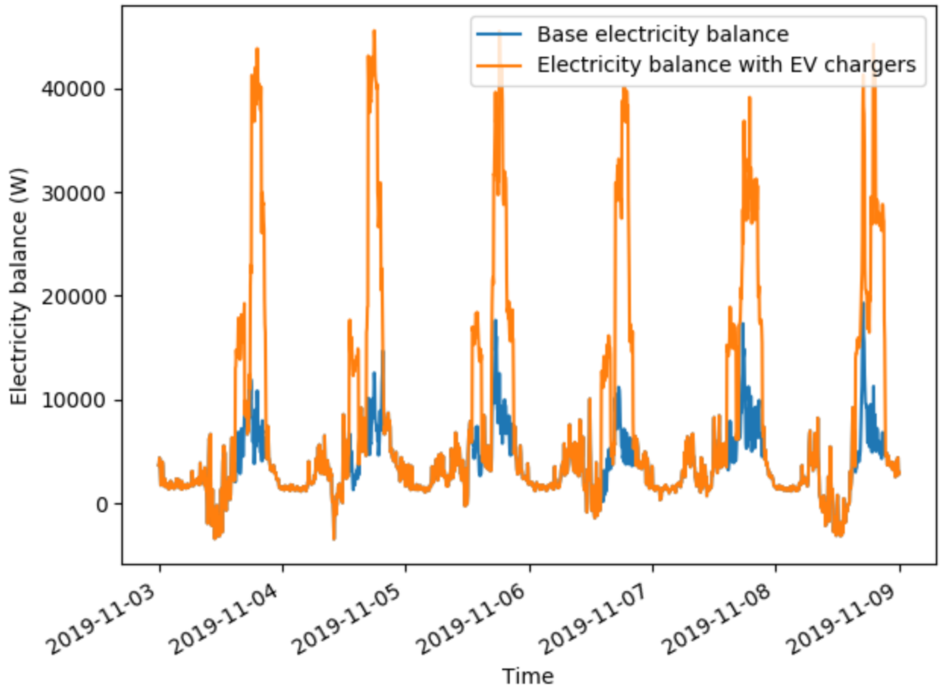

Figure 5.1: Standard simulation run with non-optimized charging

spread over the time the cars are connected to their chargers. Most charging happens at night, after the peak electricity load of the previous day.

5.2 Evaluation of goals

We have seen that our algorithm succeeds in lowering the peak load on the electricity grid with the specified testing scenario. This section will evaluate our algorithm based on the goals set in section 3.2 and will test our algorithm with some other scenarios.

5.2.1 Goal 1

The EVs should be charged fully when the (expected) departure time of the EV approaches.

During the run of the tests below, all EVs were charged entirely when their expected leaving times approached. This was our expectation as our algorithm prefers to charge cars when demand is low, but it does not enforce it. When there are no available time frames in which demand is low, cars will

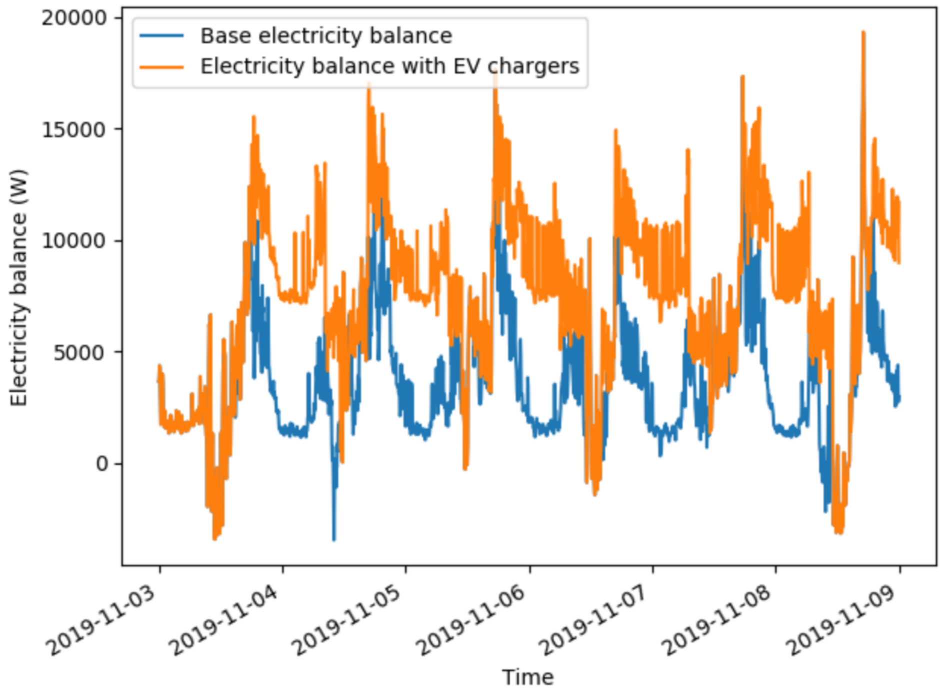

Figure 5.2: Standard simulation run with optimized charging

just charge in time frames where demand is higher. When charging a car is possible, keeping in mind the car's constraints (such as the maximum load of the charger, the connection, and leaving times of the car), our algorithm will charge it before the leaving time approaches. Goal one is thus satisfied.

5.2.2 Goal 2

The system should decrease the peak load on the electricity grid created by simultaneously charging multiple EVs.

When looking at figure 5.3, we can easily see that the peak load on the electricity grid is decreased. In table 5.1 we can see that the peak load on the electricity grid is always less than half of that when using non-optimized charging. Our algorithm almost always has a maximum of 19 308W, where the non-optimized charging maximum is around 47kW. Our algorithm has a maximum of 19 308W because the maximum peak in the test data is also 19 308W and our algorithm tries not to increase the already existing maximum peak. In comparison with the non-optimized charging algorithm, our algorithm performs very well.

When increasing the number of commuting cars, we can see that there is a

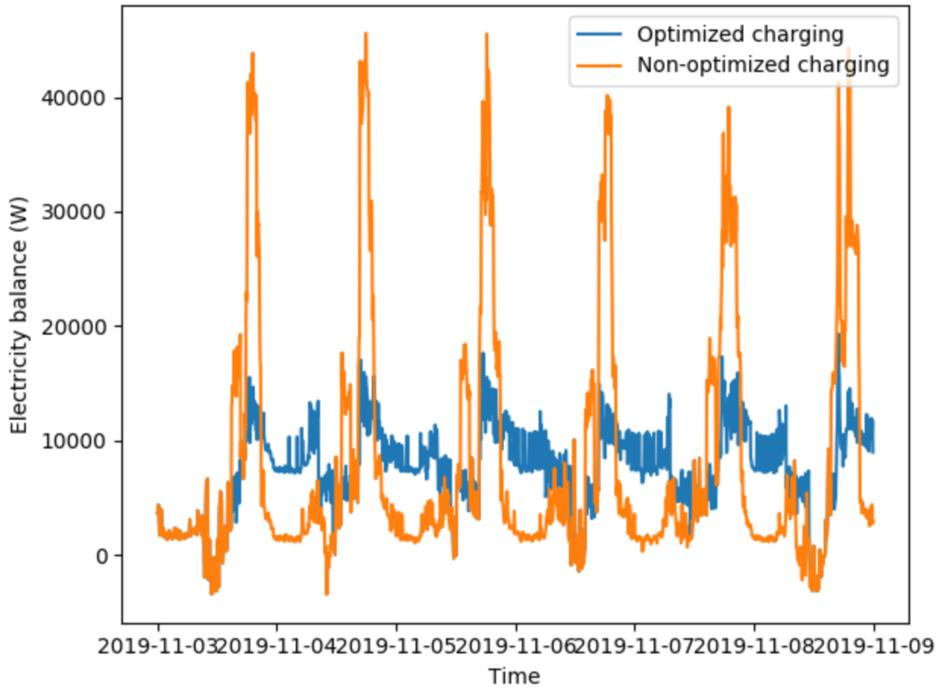

Figure 5.3: Optimized and non-optimized charging

maximum amount of cars before our algorithm starts to allocate spots above the maximum value of 19.308W in the data. The results of this test can be seen in table 5.2.

We can see that, when using our algorithm, the peak load on the electricity grid increases after four cars. The increase in peak load means that our algorithm can not divide the total load of all cars over the available time while staying below the already existing maximum peak in the test data. When not using an algorithm to divide load over the amount of time, the peak load on the electricity grid gets as high as four times the maximum in the test data (with seven cars), which again highlights the importance of regulating EV charging as with only seven cars per twelve households, the peak load becomes four times higher than it is now.

5.2.3 Goal 3

The EVs should be charged relatively fast until hitting 80% battery capacity.

We will measure the time it takes from plugging in an EV until the battery is at least 80% charged for all commuting vehicles (the vehicles that need to charge during the night). The results of this test can be seen in table 5.3.

Run	Optimized charging	Non-optimized charging
1	19 308W	45 576W
2	20 640W	50 640W
3	19 308W	49 080W
4	19 308W	45 516W
5	20 868W	50 328W

Table 5.1: Peak load on the electricity grid in normal testing scenario

Commuting cars	Optimized charging	Non-optimized charging
1	19 308W	25 688W
2	19 308W	36 688W
3	19 308W	47 688W
4	19 308W	56 576W
5	20 280W	67 576W
6	23 688W	77 280W
7	24 636W	88 280W

Table 5.2: Peak load on the electricity grid, varying the amount of commuting cars

For our proposed algorithm, the minimum time until a battery is charged to at least 80% is seven hours. If someone is to park their car at home at 18:00, the car is, on average, considered usable at 01:00 the next morning. Such long charging times are acceptable but not desirable as most cars are parked between ten and fifteen hours each night and will thus be charging a long time during these hours. As a comparison, when charging at maximum speeds, the average time to charge to 80% battery is around one hour. The proposed algorithm is thus much slower at charging the cars than uncontrolled charging as expected.

We can tweak our algorithm a bit to change the charging times of the cars. An option we have is to make the prediction window smaller. Making the prediction window smaller will force cars to charge faster as the allocation window of the cars decrease. Shrinking the prediction window also highers the chances of cars using energy at times when it is not the most optimal. The results of this test are displayed in figures 5.4 and 5.5.

We can see in figure 5.4 that the charging times of the cars start to flatten after eleven hours. We could explain this by looking at the average time the cars are parked, which is around ten to fifteen hours. When the prediction window is larger than the time a car is parked, the prediction window will

Run	Optimized charging (hours)	Non-optimized charging (hours)
1	6,79	1,08
2	8,30	1,69
3	7,66	1,19
4	6,83	0,92
5	7,97	1,58

Table 5.3: Time to charge until 80% battery capacity in normal testing scenario

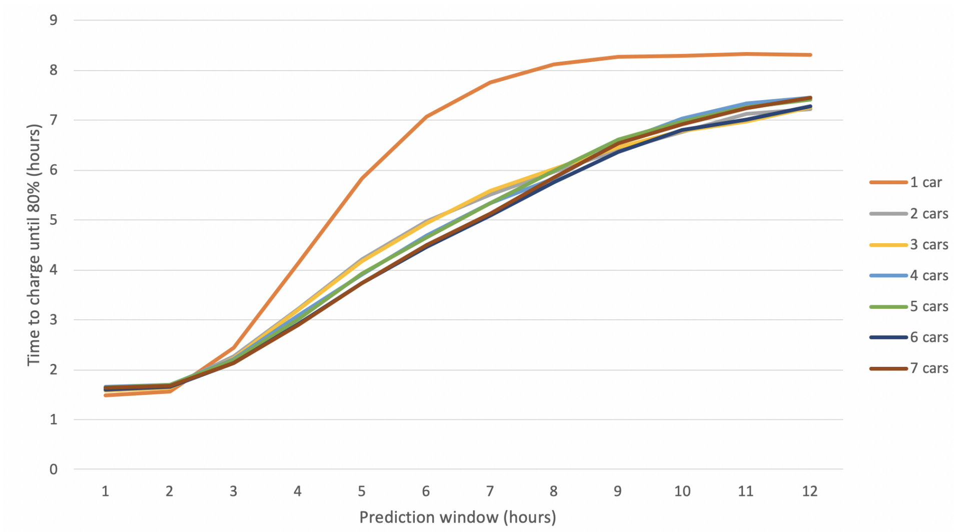

Figure 5.4: Time to charge until 80% battery capacity with variable amount of commuting cars

be adjusted to the time the car is parked, which would result in the charging times flattening, as in figure 5.4.

Another thing to notice in figure 5.4 is that the time to charge one car is different than for the other amount of cars. With only one car, charging times are a lot higher. The higher charging times are probably caused by the nature of our test data and our algorithm: with one car, we do not need to allocate much load in the test data, which results in the load being allocated in the lowest time frames in the test data. As we have seen, the electricity usage overall decreases during the night (after the peak load), which causes our algorithm to allocate much load at the end of the allocation window, making the time it takes to charge the car longer. When testing with two cars, the total load of both cars more than fills the valley after the peak load, which causes the algorithm to allocate load in time frames that are

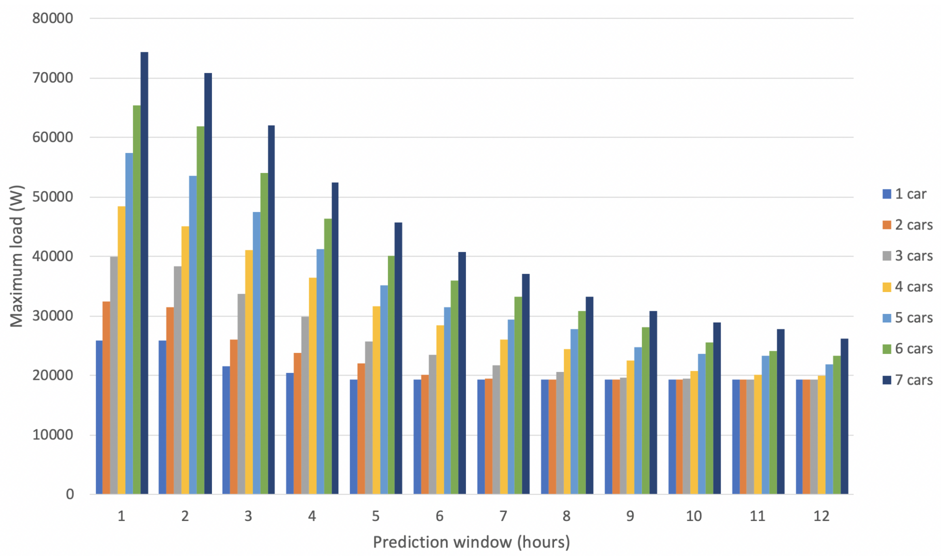

Figure 5.5: Maximum load on the electricity grid with variable amount of commuting cars

earlier in the test data, resulting in shorter charge times.

The last thing to note is that there is a balance between the time it takes to charge the cars (in figure 5.4) and the maximum load on the electricity grid (in figure 5.5). Increasing the prediction window causes a more optimal use of the available prediction data, decreasing the maximum electricity demand, but it also causes the charging times of the cars to get higher.

5.2.4 Goal 4

The system should prefer using green energy generated nearby when available.

To evaluate this goal, we first have to count how much green energy is available within the test data to later see how much of the available green energy gets used by our algorithm. We define the term green energy within our test data to be all negative energy balances, indicating a surplus of green energy. The total amount of available green energy in our test data is 11 060Wh. We ran a test with both our algorithm and non-optimized charging and counted the amount of green energy that was used. Results of the test can be seen in table 5.4.

The table shows us that the amount of green energy used by our algorithm is higher than non-optimized charging but is still rather low, only using 33%

Run	Optimized charging	Non-optimized charging
1	1 639Wh	0Wh
2	2 272Wh	0Wh
3	3 056Wh	0Wh
4	2 829Wh	0Wh
5	3 700Wh	1 304Wh

Table 5.4: Used green energy in normal testing scenario

of the available green energy in run three. To explain this, we take a look at figure 5.2 to see where the electricity balance in our data is negative (meaning more electricity is generated than is used). We can see that, most of the time, the electricity balance with EV chargers follows the baseline when more energy is generated than is used. This might be because the negative valleys (valleys below the x-axis) in our data caused by energy generation are before 12:00. As we may remember from section 4.4, there are no cars that start charging shortly before 12:00, causing many cars to be already full or not connected before the negative valleys in the test data. We predict that if some cars are connected before the negative valleys in the data, the algorithm prefers to charge these cars with the green energy available. To test our prediction, we simulated with three cars, all connecting shortly before or after the negative valleys in the data started. The results are shown in table 5.5.

Run	Optimized charging	Non-optimized charging
1	9 772Wh	9 140Wh
2	10 115Wh	9 722Wh
3	9 343Wh	9 167Wh
4	9 530Wh	9 363Wh
5	9 201Wh	9 163Wh

Table 5.5: Used green energy with cars connecting before valleys

We can see that the algorithm chooses to charge the cars with green energy in this particular scenario. This is a more than three times improvement in comparison with the runs in table 5.4. Also, the non-optimized runs are vastly improved. We may thus conclude that the connection times of cars maybe matter more than how our algorithm works. In both cases, the connection times matter very much.

The above conclusion might not be what we expected because our algorithm seeks the most optimal times to charge. However, due to the nature of our testing data, not much solar energy is available to test our algorithm. For example, if our data would have longer negative valleys, we would expect to see that our algorithm would choose to charge the cars during the entire time of the valley. A run without our algorithm would just charge fast at the beginning of the valley until the car is full. The data we used has too short valleys to prove our claim. The peak caused by normal charging is as big or bigger than the length of the valleys in the data, causing the gap in this test to be very small. More on this will be discussed in section 7.4.

The last test we ran considered the amount of green energy used when changing the prediction window, just as in goal three. We hypothesize that the amount of green energy used for charging increases when the prediction window of the algorithm gets larger. The algorithm can then see negative valleys that might lie further in the future and allocate car charging to these valleys. The results of the test are shown in table 5.6.

Prediction window (hours)	Green energy usage
1	0Wh
2	0Wh
3	0Wh
4	73Wh
5	255Wh
6	403Wh
7	988Wh
8	1 567Wh
9	1 759Wh
10	1 759Wh
11	1 759Wh
12	1 759Wh

Table 5.6: Amount of green energy used in the normal testing scenario with a variable prediction window

As expected, we see that the amount of green energy used by our algorithm increases when the prediction window gets larger.

We have seen that our algorithm uses more green energy than uncontrolled charging and thus achieves goal four. However, in the second test of this section, the results show that connection times of EVs matter a lot in the usage of green energy. We might argue that this is due to the nature of our testing data (not a lot of solar power is generated in the autumn). It would be nice if more testing were done with data from the summer because of the higher amounts of generated solar energy in the summer.

5.2.5 Summary

There are a couple of things we can conclude from the tests ran above:

- When making the prediction window bigger, the time it takes the cars to charge until they are 80% full increases.
- When making the prediction window bigger, the amount of green energy used increases.
- When making the prediction window bigger, the peak load on the electricity grid decreases.

The problem with the goals is that compliance to all of them is hard as with only four cars and a prediction window of twelve hours, the peak load is already increased. There is thus no option in which we choose a prediction window that complies with all the goals. We need to make a trade-off between our goals to decide the best possible prediction window.

We can not choose a universal setting for the prediction window in the algorithm as the baseload, amount of cars and the amount of generated green energy in a neighborhood all highly influence the above conclusions. For example, when more energy is generated it might be a better choice to higher the average charging time of the EVs a bit and to use more green energy. The optimal prediction window also highly depends on the amount of commuting and non-commuting vehicles in a neighborhood and when these vehicles are used. If cars connect long before green energy is produced, a lower prediction window might cause cars to not charge in times where green energy is available, but when cars tend to connect shortly before green energy is produced, cars could benefit from a shorter prediction window.

To what length the prediction window needs to be set should thus be decided on a per neighborhood basis. We might be able to distinguish patterns between the amount of EVs in a neighborhood, solar panels in a neighborhood, and the optimal prediction window, but in order to recognize these patterns, further testing is necessary.

Chapter 6

Related Work

The impact of EV charging is thoroughly described by Gómez et al. [9] and Clement-Nyns et al. [3]. Clement-Nyns et al. also describe a method of balancing the grid impact by using quadratic programming. The main distinction can be made by papers that use quadratic programming methods to smartly charge EVs and other methods to charge EVs smartly. Some authors that use other methods for smartly charging EVs are Mets et al. [17], Sortomme et al. [24] and van Dael et al. [27].

A lot of papers use the same methods as we do [3][17][27]. These methods assume that we know the leaving times of the EVs, know a prediction of the household load, and can control cars as we do.

Mets et al. propose two methods for optimized EV charging in their paper. A local method that only optimizes the load of one household and a global method that optimizes the load of a residential area. Vandael et al. [27] propose a different method of decentralized EV management in which they use low and high voltage transformer regulators to achieve an optimal load.

Another method of optimal charging is deciding whether to charge an EV based on current electricity pricing (price-signaled charging). An example of such an algorithm has been proposed by Markel et al. [15]. Charge optimization based on electricity pricing is most of the time not that different from regular load-based optimization as the electricity price is often influenced by the load on the electricity grid.

A different type of EV charging algorithm uses vehicle to grid (V2G) capabilities to regulate the load on the electricity grid by not only smartly charging EVs but also smartly discharging them [13]. Such an algorithm can be found in a paper by Mets et al. [16].

Vehicle to grid services can put extra load on EV batteries and can cause those batteries to fail earlier than expected. An evaluation of EV batteries and vehicle to grid services can be found in papers by Bishop et al. [2] and Lunz et al. [14].

Another thing that stands out from the literature is the fact that many papers use PHEVs (plug-in hybrid electric vehicles) instead of EVs [17][8]. This might be because PHEVs were more common than EVs when these papers were created. An advantage of using PHEVs is that the usability aspect we evaluated in this paper is much less relevant because when owners need their PHEV unexpectedly, they could always use the non-electric fuel source in their PHEV.

Chapter 7

Future work

This chapter will go over topics that were not discussed in this paper but have come up during the making of it. We will discuss the possible implementation of bidirectional charging in section 7.1. Privacy issues of storing electricity usage data are discussed in section 7.2. Possible means of how to capture the needed data for our algorithm is discussed in section 7.3. Section 7.4 will elaborate upon the test data we used and the test data used by other similar research. Section 7.5 will propose some improvements to our algorithm.

7.1 Bidirectional charging

At the beginning of this paper, we noted that bidirectional charging is in development and has the potential of being able to reduce the load on the power grid even more than algorithms that only use unidirectional charging. Normally, when electricity demand is high, electricity production plants scale up their production so that more energy is provided to the electricity grid. With bidirectional charging, this might not be needed as much because EVs can discharge on-demand. When demand for electricity is high, EVs within a neighborhood can start discharging to provide the neighborhood with more electricity. The discharged EVs would be able to recharge their batteries when demand is low again. As an example, a method for using bidirectional charging to achieve peak-shaving is proposed in a paper by Wang et al. [29].

7.2 Privacy impact of fine-grained P1 data

As we have stated in the previous chapters, for our algorithm to work, we need an aggregated set of electricity usage and generation data from neighborhoods. The data can be collected from smart meters, just as Wijkie does for their algorithm. However, it can be argued that the processing of consumer electricity data is privacy sensitive. This is also confirmed by Cuijpers et al. [4] in a paper about the enrollment of Dutch smart electricity meters. It was confirmed that the enrollment of smart electricity meters that send usage data to CPO's violated the Dutch Data Protection Act (Wet bescherming persoonsgegevens) and article 8 of the ECHR (European Convention for the Protection of Human Rights and Fundamental Freedoms) because electricity usage data was considered to be personal data. Electricity usage data could be used to derive information about lifestyle, presence or absence and the number of people in a house and is thus considered personal data. Companies processing electricity usage data should thus also have to comply with the GDPR.

Moreover, there might be ways in which electricity usage data could be seen as sensitive personal data, causing companies that process this data to make a privacy impact assessment (or PIA for short). Two examples are explained below.

According to a report on smart electricity meter data from the US Congressional Research Service [18], there are two major privacy impacts associated with smart electricity meter data, one of which can be applied to the way we use the electricity data. The paper states that electricity usage data of a house can be used to determine a couple of things about the people living in that house, such as what electrical appliances they have and their daily schedule. Even anonymous data can sometimes be traced back to an individual or a household by using publicly available information [18]. According to the report, electricity usage data could also be used to predict what kind of medical devices a household uses, linking electricity data to the physical health of a resident. Data that could be used to identify the mental or physical health of a person is considered sensitive personal data under the GDPR. If a company processes sensitive personal data, it is obliged to make a PIA and access the privacy risks that could occur when some of their data is stolen.

Another example of how electricity usage data can be used to derive other facts about a person has to do with religion. Orthodox Jews are not allowed to operate electrical devices on Shabbat (day of rest, Saturday) [10]. However, they are allowed to preset their electrical devices before Shabbat so that they do have lights turned on during Shabbat. This might be reflected in their electricity usage data, meaning that the electricity usage data could be used to derive that a household has Jewish residents. Religion is also considered sensitive personal data under the GDPR and has the same effect as mental or physical health data. A company processing data that could be linked to religion is thus also obliged to make a PIA under the GDPR.

Though the examples above may be considered extreme, they might become an issue in the future. Advice to companies (including Wijkie) that are using energy usage data is thus to at least perform a PIA to meet the demands stated in the GDPR.

7.3 Data gathering from EVs

In section 3.3.2 we specified the input data for our algorithm. The input data for our algorithm included a lot of data that had to be gathered from the EVs. To recap, the data gathered from the EVs is the following:

- C_e^{total} for all EVs e in the neighborhood connected to a charging pole, being the total capacity of a car.
- C_e^{now} for all EVs e in the neighborhood connected to a charging pole, being the current capacity of a car.
- t_e for all EVs e in the neighborhood connected to a charging pole, being the leaving time of EV e.
- L_e^{max} for all EVs e in the neighborhood connected to a charging pole, being the maximum load of the charger and the car.

In order to get this information to Wijkie's servers, we need a protocol that supports transferring this information from the charge points and electric vehicles. Protocols that look promising for supplying C_e^{total}, C_e^{now} and L_e^{max} are Mode 3 and OCPP. OCPP can even be expanded in functionality to comply by the using party's requirements. Mode 3 could be used to provide communication between EV and charge point (see the example in figure 7.1[1]).

Furthermore, a choice has to be made on how to get data about the leaving times of EVs (t_e). Wijkie could, for example, decide to expand their smartphone application in order to give the ability to their users to set leaving times or schedules for their EV usage. Another opportunity lies in the prediction of t_e as users might not like entering their leaving times within an app every day. Smartphones already do something similar predicting when they will be needed and changing their charging habits based on that prediction [1].

[1] Figure 7.1 displays Wijkie as CPO.

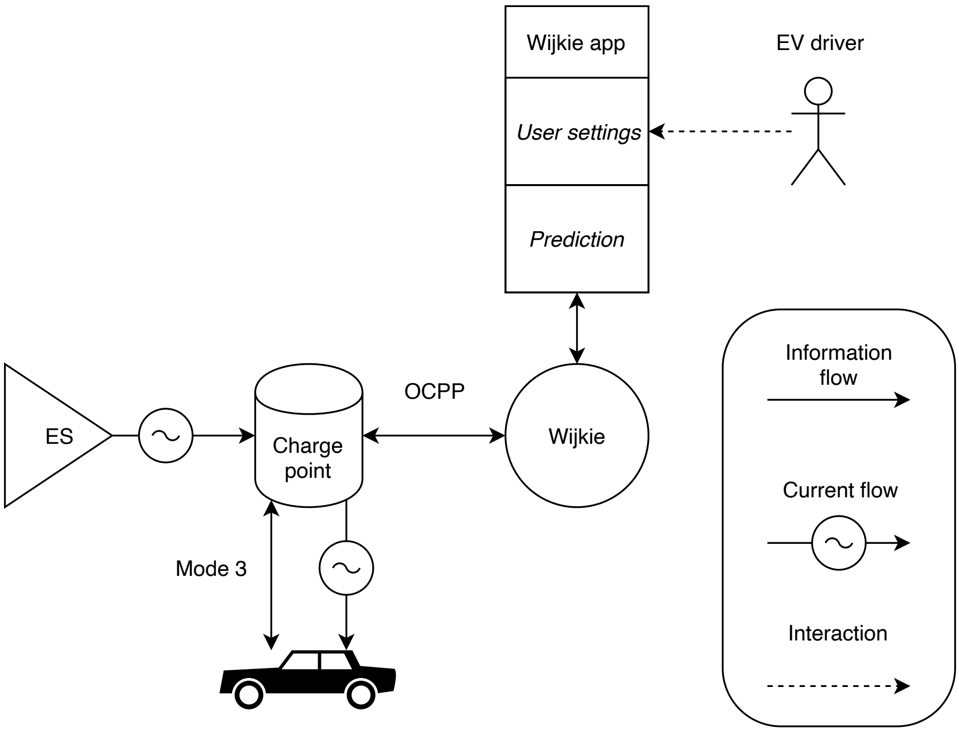

Figure 7.1: Proposed usage of protocols

A combination could be made where Wijkie's app predicts leaving times of the EVs, but these leaving times can be overwritten by the users (displayed in figure 7.1). More research is needed for deciding which methods to use to gain the required information.

7.4 Test data

A lot of studies about smart EV charging use synthetic load profiles for testing purposes [3][17][24][27]. Synthetic load profiles are made by electricity suppliers to predict the energy usage on specific time frames. These load profiles are an average over the consumer base of an electricity supplier. Electricity suppliers adjust their electricity production based on these predictions. The usage of synthetic load profiles has advantages and disadvantages. It mainly depends on how many houses the electricity usage data should include. When using entire cities, synthetic load profiles are probably a good estimation of the total electricity usage. When simulating the usage of only a couple of households, synthetic load profiles might not be accurate enough.

During the making of this paper, we used aggregated data from Wijkie because real electricity usage data with a short time interval was not easy to find. The topic of smart EV charging needs a lot more research in the upcoming years. Real electricity usage data is needed to make more accurate tests for the developed algorithms. Electricity providers and other parties should provide aggregated electricity usage data to be used for the development of EV charging algorithms.

Because we did not have a lot of test data available, we were only able to test our algorithm based on electricity data from a week in November. As we already stated in section 5.2.4, testing our algorithm with data from different seasons might bring new insights on how our algorithm performs.

7.5 Algorithm improvements

Our proposed algorithm might be improved in future research. We listed a couple of points we think our algorithm could profit from.

7.5.1 Removing the minimum allocation unit

We used a minimum allocation unit (AU^{min}) to allocate charging blocks for the EVs. We might be able to define our algorithm without using the minimum allocation unit by using the differences between the minimal time frames. Figure 7.2 displays an example.

Instead of using the minimum allocation unit for allocating fixed chunks of charging spots, we allocate charging spots based on the minimal and next after minimal time frame (**1** in figure 7.2). This repeats itself until the car is completely allocated (**2** in figure 7.2). Not using the minimum allocation unit might let our algorithm allocate EV charging more smoothly.

7.5.2 Better fast charging

Another improvement for our algorithm might be implementing a way to influence partial car charging. We have shown that by making the prediction window smaller, cars are charged faster, but this was an indirect consequence of the smaller prediction window. With this method, we are not able to guarantee that cars are always 80% charged x hours after they are plugged in. A possible way to guarantee that cars are charged earlier is to change the way they charge, using another charging strategy for the first 80%. Smartly doing this might help with decreasing the charging times and will set a guarantee that cars are partially charged at a specific moment in time.

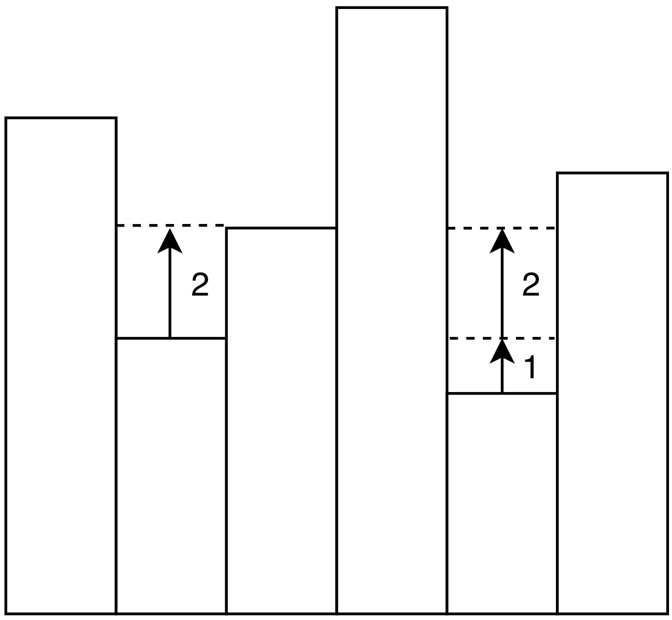

Figure 7.2: Proposed algorithm improvement for removing the minimum allocation unit

7.5.3 Predicting the connection time of EVs

Our algorithm might also benefit from a prediction of the connection time of EVs. Knowing beforehand when and how many EVs will be connecting to the power grid could result in our algorithm scheduling EV charging more optimally.

7.5.4 Changing the allocation order

The last proposed improvement is a clever trick that might improve the algorithm and lower the maximum peak. When allocating charging for a vehicle and two time frames have the same minimal load, it will always prefer the earliest time frame for charging. An example is displayed in figure 7.3.

In the left graph, car $e2$ gets allocated first. Like all cars, it prefers to fill the earliest time frame if there is more than one minimal time frame. Car $e2$ thus prefers time frames one and two over three and four. After car $e2$'s allocation, car $e1$ gets allocated but has a smaller allocation space and can thus only allocate in time frames one and two. It does so on top of the already made allocation by vehicle $e2$ resulting in a higher peak demand than in the right graph.

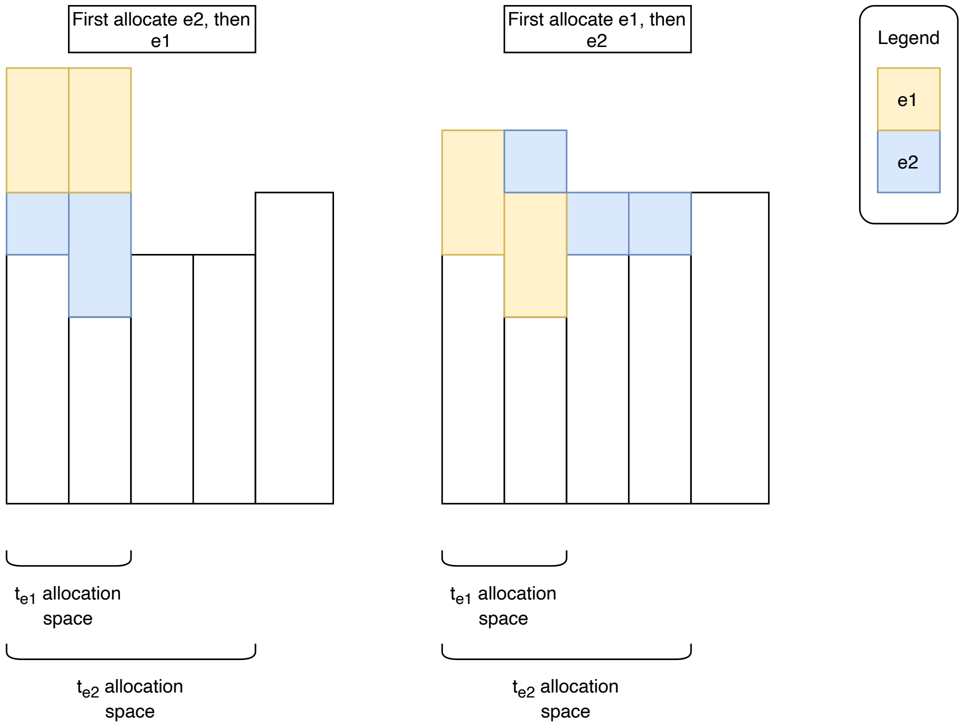

Figure 7.3: Proposed algorithm improvement for optimizing the allocation order

The right figure displays the allocation the other way around. Car $e1$ gets allocated first and allocates itself in time frames one and two. Car $e2$ will now allocate in time frames three and four because these time frames are now minimal. The result is a lower maximum peak than in the left graph.

Allocating the most restricted vehicle first (the vehicle with the smallest allocation window) might result in lower maximum peaks and more clever allocation, as seen in the example. Our algorithm used an arbitrary order in which to allocate EV charging. Sorting the EVs before allocation might help achieve lower peak loads.

Chapter 8

Conclusions

Smart charging of electric vehicles is a complex topic. We have seen that the current household electricity usage is far less than what EV chargers are predicted to use. With the rising number of electric vehicles, the working of the current electricity grid has to change in order to prevent it from becoming overloaded. To this end, many organizations are experimenting with ways to implement smart car chargers in the current infrastructure. Protocols like OSCP and OCPP will be much needed in the future to facilitate a smarter charging infrastructure.

We have shown that an algorithm based on usage prediction can help to lower the peak load on the electricity grid. Our algorithm was tested by using the four goals explained in section 3.2. We have shown that the algorithm is capable of allocating the total load of the EVs in times where demand is lowest. We have also shown that, when using our algorithm, the peak load on the electricity grid is much lower compared to uncontrolled charging.

We also experimented with the usage of green energy by our algorithm and some problems were encountered with the nature of our test data. We manipulated the connection time of some of the vehicles to show the effect it had on green energy usage. More research with different electricity usage data will be needed to show that implementing our algorithm leads to an increased usage of locally produced green energy. We only tested with data in one season (namely a week in November). It would be best to test our algorithm with data from different seasons, as electricity usage tends to vary between seasons. For example, electricity generation from solar panels is much higher in the summer.

Our algorithm was initially configured to use a prediction window of twelve hours (see section 3.3.1 for an explanation of the prediction window and allocation space). The algorithm could thus maximally use twelve hours of prediction data for allocating EV charging. We saw that this setting caused the charging of the EVs to happen at the end of their allocation windows (because the load was lowest there). This caused charging times of the EVs to be rather long. After changing the prediction window, we saw that the charging times of the EVs became shorter. A downside was that the maximum peak increased and green energy usage decreased. We thus need to make a trade-off between charging times, maximum load, and that green energy usage.

The trade-off we must make should be made on a per neighborhood basis as the following influence the peak load and green energy usage:

- The number of EVs within a neighborhood: more EVs means more load, which means the peak load could be potentially lowered by making the prediction window longer.
- The usage patterns of EVs within a neighborhood: when a lot of EVs connect long before the production of green energy, the neighborhood could benefit from a longer prediction window, but when EVs connect short before the production of green energy, other EVs could benefit from a shorter prediction window.

All with all, choosing the right prediction window should be done after more testing. Maybe a pattern could be distinguished based on the points made above, or the prediction window could be changed while running the algorithm. However, more testing is necessary for evaluating and fine-tuning our algorithm.

Bibliography

[1] Apple. About Optimized Battery Charging on your iPhone. `https://support.apple.com/en-us/HT210512`.

[2] J. D. Bishop, C. J. Axon, D. Bonilla, M. Tran, D. Banister, and M. D. McCulloch. Evaluating the impact of V2G services on the degradation of batteries in PHEV and EV. *Applied energy*, 111:206–218, 2013.

[3] K. Clement-Nyns, E. Haesen, and J. Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on power systems*, 25(1):371–380, 2009.

[4] C. Cuijpers and B.-J. Koops. Smart metering and privacy in Europe: lessons from the Dutch case. In *European data protection: Coming of Age*, pages 269–293. Springer, 2012.

[5] T. DiChristopher. Electric vehicles will grow from 3 million to 125 million by 2030, International Energy Agency forecasts, May 2018.

[6] M. Dubarry, A. Devie, and K. McKenzie. Durability and reliability of electric vehicle batteries under electric utility grid operations: Bidirectional charging impact analysis. *Journal of Power Sources*, 358:39–49, 2017.

[7] Edmunds. 2019 Tesla Model S Features & Specs. `https://www.edmunds.com/tesla/model-s/2019/features-specs/`.

[8] J. García-Villalobos, I. Zamora, J. San Martín, F. Asensio, and V. Aperribay. Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches. *Renewable and Sustainable Energy Reviews*, 38:717–731, 2014.

[9] J. C. Gómez and M. M. Morcos. Impact of EV battery chargers on the power quality of distribution systems. *IEEE Transactions on Power Delivery*, 18(3):975–981, 2003.

[10] B. Greenberg. *How to run a traditional Jewish household.* Simon and Schuster, 2011.

[11] J. Hosking, R. Natarajan, S. Ghosh, S. Subramanian, and X. Zhang. Short-term forecasting of the daily load curve for residential electricity usage in the Smart Grid. *Applied Stochastic Models in Business and Industry*, 29(6):604–620, 2013.

[12] C. S. Ioakimidis, D. Thomas, P. Rycerski, and K. N. Genikomsakis. Peak shaving and valley filling of power consumption profile in non-residential buildings using an electric vehicle parking lot. *Energy*, 148:148–158, 2018.

[13] W. Kempton and J. Tomić. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of power sources*, 144(1):280–294, 2005.

[14] B. Lunz, Z. Yan, J. B. Gerschler, and D. U. Sauer. Influence of plug-in hybrid electric vehicle charging strategies on charging and battery degradation costs. *Energy Policy*, 46:511–519, 2012.

[15] T. Markel, M. Kuss, and P. Denholm. Communication and control of electric drive vehicles supporting renewables. In *2009 IEEE Vehicle Power and Propulsion Conference*, pages 27–34. IEEE, 2009.

[16] K. Mets, T. Verschueren, F. De Turck, and C. Develder. Exploiting V2G to optimize residential energy consumption with electrical vehicle (dis) charging. In *2011 IEEE first international workshop on smart grid modeling and simulation (SGMS)*, pages 7–12. IEEE, 2011.

[17] K. Mets, T. Verschueren, W. Haerick, C. Develder, and F. De Turck. Optimizing smart energy control strategies for plug-in hybrid electric vehicle charging. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 293–299. IEEE, 2010.

[18] B. J. Murrill, E. C. Liu, and R. M. Thompson. Smart meter data: Privacy and cybersecurity, 2012. Congressional Research Service Washington, DC.

[19] Open Charge Alliance. Open Charge Point Protocol. Protocol specification. `https://www.openchargealliance.org/protocols/ocpp-20/`.

[20] C. M. Portela, P. Klapwijk, L. Verheijen, H. De Boer, H. Slootweg, and M. Van Eekelen. OSCP-an open protocol for smart charging of electric vehicles. In *Proc. 23rd Int. Conf. Electr. Distrib.(CIRED)*, pages 1–5, 2015.

[21] K. M. Powell, A. Sriprasad, W. J. Cole, and T. F. Edgar. Heating, cooling, and electrical load forecasting for a large-scale district energy system. *Energy*, 74:877–885, 2014.

[22] G. A. Putrus, P. Suwanapingkarl, D. Johnston, E. C. Bentley, and M. Narayana. Impact of electric vehicles on power distribution networks. In *2009 IEEE Vehicle Power and Propulsion Conference*, pages 827–831, Sep. 2009.

[23] C. Sandels, J. Widén, L. Nordström, and E. Andersson. Day-ahead predictions of electricity consumption in a Swedish office building from weather, occupancy, and temporal data. *Energy and Buildings*, 108:279–290, 2015.

[24] E. Sortomme, M. M. Hindi, S. J. MacPherson, and S. Venkata. Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses. *IEEE transactions on smart grid*, 2(1):198–205, 2010.

[25] M. Turcotte. Commuting to work: Results of the 2010 general social survey. *Canadian Social Trends*, 92(August):25–36, 2011.

[26] P. Van Aubel and E. Poll. Communication Security of EV-Charging Protocols. 2019.

[27] S. Vandael, N. Boucké, T. Holvoet, and G. Deconinck. Decentralized demand side management of plug-in hybrid vehicles in a smart grid. In *Proceedings of the first international workshop on agent technologies for energy systems (ATES 2010)*, pages 67–74, 2010.

[28] Vlaamse Regulator van de Elektriciteits- en Gasmarkt. De verbruiksprofielen van huishoudelijke en niethuishoudelijke elektriciteitsverbruikers voor het jaar 2019. `https://www.vreg.be/nl/verbruiksprofielen`.

[29] Z. Wang and S. Wang. Grid power peak shaving and valley filling using vehicle-to-grid systems. *IEEE Transactions on power delivery*, 28(3):1822–1829, 2013.

[30] Y. Wi, J. Lee, and S. Joo. Electric vehicle charging method for smart homes/buildings with a photovoltaic system. *IEEE Transactions on Consumer Electronics*, 59(2):323–328, May 2013.