Implementation of a parking choice model to analyse the effect of parking prices on electric vehicles

Valentin Bemetz
Moritz Hohenfellner

Supervisors:
Prof. Dr. Kay W. Axhausen
Dr. Rashid A. Waraich

Bachelor thesis

Institute for Transport Planning and Systems **June 2014**

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

List of Figures

List of Tables

Photo credit cover page:
Rocky Mountain Parking Lot Services, Inc. (http://www.parkinglotservices.com/recent-jobs)

Bachelor thesis

Implementation of a parking choice model to analyse the effect of parking prices on electric vehicles

Valentin Bemetz, Moritz Hohenfellner
IVT
ETH Zürich

CH-8093 Zürich

valentin.bemetz@gmail.com,
moritz@hohenfellner.de

June 2014

Abstract

The increasing market penetration of electric vehicles makes it inevitable to consider their specific behaviour in terms of stationary traffic. It has to be investigated where charging infrastructure should be placed, which kind of charging infrastructure should be provided and how to make optimum use of the existing charging infrastructure by changing policies.
This thesis presents a parking choice model which considers both conventional and electric vehicles. Therefore an existing model was extended by two additional variables to consider the charging behaviour of electric vehicles. Further the integration of the model into the agent based transport simulation framework MATSim is documented.
Additionally an example for the application of the model in policy studies is given using the enriched Sioux Falls scenario. The required integration of parking and charging infrastructure into the scenario as well as further adaptations are described in this work. The evaluation of the experiments showed that the aim to make electric vehicle driving more attractive by reducing walking distances could be accomplished by modified policies.

Keywords
Parking choice, Electric vehicles, Agent based simulation, MATSim, Sioux Falls scenario

Preferred citation style

Bemetz, V. and Hohenfellner, M. (2014) Implementation of a parking choice model to analyse the effect of parking prices on electric vehicles, *Bachelor thesis*, ETH Zurich, Zurich.

Acknowledgements

We would like to thank Prof. Dr. Axhausen for giving us the chance to work on this challenging but very exciting project and for leaving us to move freely within a broadly defined task. His inputs were always furthering this work.

Further we would like to thank Dr. Rashid A. Waraich for supervising this work and for always taking time for our short notice meetings.

Finally we would like to thank Artem Chakirov for providing us the data of the Sioux Falls scenario and giving us helpful advice.

1 Introduction

1.1 Background

1.1.1 MATSim

Multi agent transport simulation (MATSim) is an agent based transportation simulation framework implemented in Java. MATSim generally needs three types of input data (MATSim, 2014).

- The network defines links, nodes, capacities etc.
- Facilities: The agents can perform activities at facilities. Each facility has a location and activity types which can be performed at it, for example work or education. Additionally facilities can have specific opening hours and capacities.
- The population defines each agent with his specific plans and attributes. The plans contain the activities an agent wants to perform during one day, the facilities where he wants to perform them at and the legs which define how he gets from one activity to another.

During the simulation each agent tries to perform his selected plan. After an iteration, which typically corresponds to one day, each agent receives a score for his plan. The score basically takes into account how much time the agent had to perform activities and how much time he had to spend on travelling. This score is calculated by the utility function, which can be affected by additional information for example disutility for travel related expenses during the day or delay caused by congestion.
Between two iterations agents can try to improve their scores by changing their plans. For example an agent can decide to go to work earlier to avoid the rush hour or to change his travel mode. Each plan and the score the agent reached performing it is saved. At the end of the simulation each agent should have maximised his own score by sorting out bad plans.
MATSim is a very powerful tool but does not per default take account of stationary traffic. Without additional plug-ins a car disappears from the network when an agent reaches his destination.

1.1.2 Existing parking choice model

Waraich and Axhausen (2012) presented a parking choice model which is implemented as an add-on to MATSim and applied it on the city of Zurich to optimise parking prices (Waraich

et al., 2012). In that model a parking score is assigned to each available parking within a given distance. The parking which has the highest score is chosen by the agent. If there is no available parking within the default distance, the distance is increased until the agent is able to park.
The utility function used to rate each parking takes the fees and the distance an agent has to walk into account.

$$U = \sum x_i \cdot \beta_i \tag{1}$$

As shown in equation 1 these items are weighted by β parameters, which can differ from agent to agent for example depending on their income. The influence of the walking distance is assumed to be two phase linear, as shown in equation 2.

$$U_{walking}(x) = \begin{cases} a \cdot x & x < v \\ a \cdot x + \beta \cdot (x - v) & x \geqslant v \end{cases}, \qquad \alpha, \beta < 0; \alpha >> \beta \tag{2}$$

That model only provides limited support for applications related to the parking choice of electric vehicles.

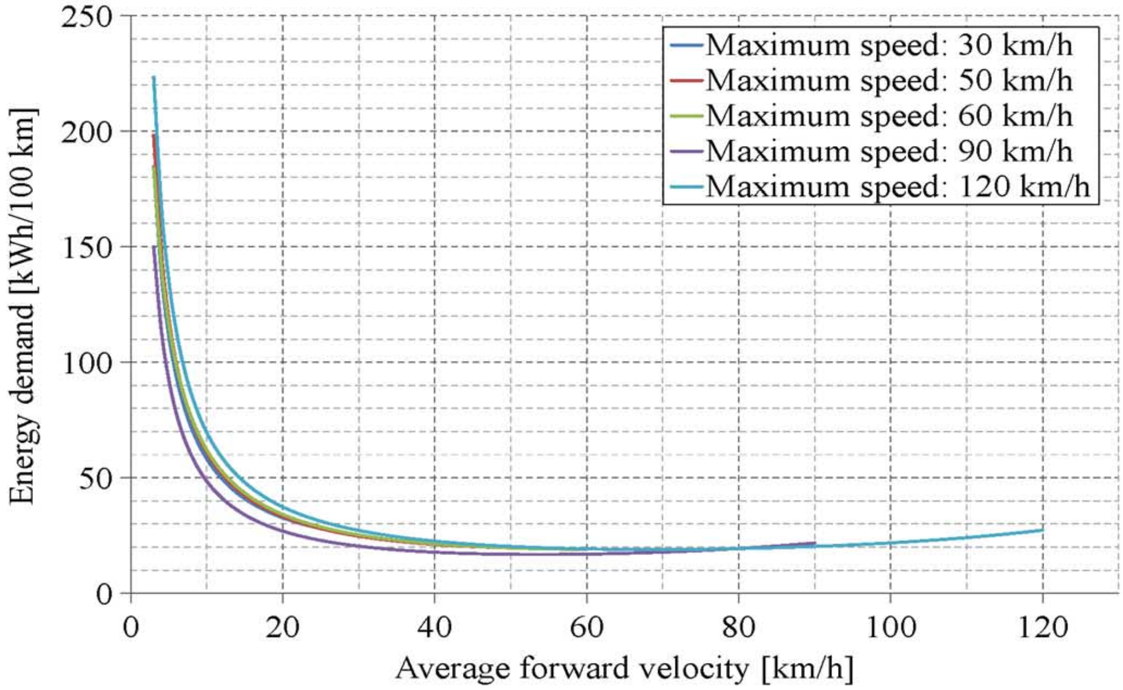
1.1.3 Existing electric vehicle models

Galus et al. (2012b) presented a model which combines the evolution of the composition of the future vehicle fleet, regarding different types of power trains, with energy consumption models for different vehicle types.
The evolution of the future vehicle fleet is simulated based on the current market shares and of a survival probability of the vehicles, which describes how likely it is that a car is replaced in a specific year.
The energy consumption models consider different combinations of several design parameters such as power train type and weight. To calculate the energy consumption for a given leg, the maximum possible speed of the links and the agent's actual average velocities are required. The parameterized energy demand is illustrated in figure 1.

1.2 Previous work

Previous work focused on the analysis of the impact of electric vehicles on the electric grid, for example Galus et al. (2012a,b). Some of those analyses require a parking choice model to describe the physical limits of parking capacity. Since they focus on the electric grid they do not

Figure 1: Average energy consumption of electric vehicles in dependency of the possible maximum speed of the link and the actual velocity driven.



Source: Galus et al. (2012b)

consider the distribution of charging infrastructure on a single parking space scale but assume that there is enough charging infrastructure available (Galus et al., 2012a) or that every parking space belonging to a specific activity type, for example home or work, is equipped with charging infrastructure (Galus et al., 2012b).

Schieffer (2010) and Galus (2012) investigated different approaches for smart charging of electric vehicles, for example, dependent on the electricity rate. They both assume that the vehicles are always connected to the electric grid during the time an agent performs an activity at any location, so that it is always possible to charge during the parking period and the agents are able to start and stop charging at any time.

There is no model known to us where the agents have to gauge up between taking a space with charging infrastructure and taking a space without charging infrastructure.

1.3 Motivation

With an increasing dissemination of electric vehicles it becomes more and more relevant to take the special needs of electric vehicles into account when it comes to the development of parking

policies or to decisions about the installation of new charging infrastructure.
To be able to predict the effect of such measures, existing models have to be extended by a detailed parking choice model taking also electric vehicles into account.
One possible application, which will be demonstrated in this work, is to optimise the usage of existing charging infrastructure by changing parking policies. This is an easy way to make electric vehicle driving more attractive and less limiting without installing additional, expensive electric vehicle charging infrastructure.

2 The parking choice model

2.1 Infrastructure

The parking infrastructure in this simulation consists of public and private parking lots. Private parking lots are always assigned to a facility. The usage of a private parking is only permitted to agents performing an activity at the corresponding facility.
Each parking space can be defined to be equipped with a specific type of charging infrastructure. The power provided by the charging infrastructure, and therefore the time it takes to charge, differs by type.

2.2 Conventional utility function

During the parking choice process an agent has to select a parking space out of a previously generated set of available spaces. The generation of that set is described in section 3.3.1.
To select the most convenient parking space for a conventional vehicle the approach proposed by Waraich and Axhausen (2012) described above is used.
It is assumed that every agent has perfect knowledge about all available parking spaces within a given distance and always chooses the one with the highest score.
Differing from Waraich and Axhausen (2012) the income is not taken into account yet because it is not used by MATSim, weighting parking expenses by the income would therefore not be consistent with other parts of the simulation (Rieser et al., 2014). Another difference is the weighting of the walking distance, which in this case is assumed to be constant to keep the scenario consistent. Those details should be adapted to the boundary conditions of the scenario in each specific case. The utility function which is used for conventional vehicles in this work is

described by equation 3.

$$U_{space} = \beta_{walk} \cdot wdistance - \beta_{money} \cdot cost, \qquad \beta_{walk} < 0;\ \beta_{money} > 0 \tag{3}$$

2.3 Extended utility function for electric vehicles

Because the behaviour of agents driving an electric vehicle (EV) can not be described by the conventional utility function two additional items were added to it. In general there is no empirical data or previous work on this subject known to us which is suitable for this application. Therefore the model consists of assumptions, which will be discussed later on.

2.3.1 Influence of the remaining state of charge

It is crucial to take the remaining state of charge, an agent will have at his next destination, into account, especially if the state of charge (SOC) is too low to reach the next destination.

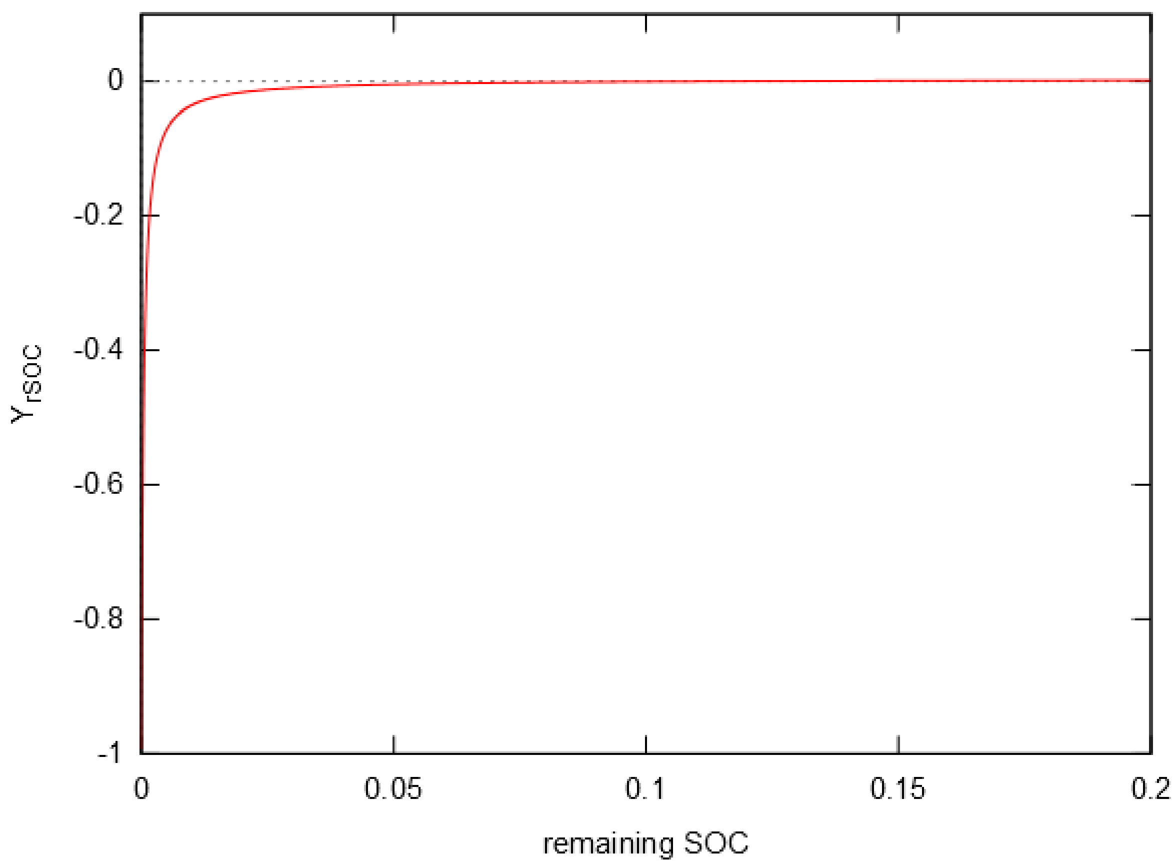
$$\text{RemainingSOC} = \text{SOC} - \text{NeededBatteryDuringRestOfDay} \tag{4}$$

Actually the charge needed to reach the agent's next destination, the charge needed to drive all of the outstanding legs of the agent's plan for the day and the possibilities to charge while performing later activities should be considered.

This would require the agents to have experience on the probability to get a parking space equipped with charging infrastructure in the area of all of the facilities they will perform activities at under consideration of the corresponding time. Additionally parking would have to be integrated into the agent's plan, to define where an agent plans to charge. Furthermore within day replanning would have to be used if an agent plans to park but there is no parking space available that provides charging infrastructure. Due to the described complexities the model does not regard the possibility of charging later during the day yet. This will only affect agents who perform more than two legs during the day which, for the scenario used in this work is not the case. Nevertheless this has to be enhanced in future work to be able to handle more complex plans in a proper way.

We assume that the remaining state of charge the agent will have at his next destination becomes seriously relevant if it is lower then 5 percent. The benefit of the remaining SOC is defined by

Figure 2: Scaling function for the remaining SOC



equation 5, which is plotted in figure 2.

$$Y_{rSOC} = \frac{\left(\left(e^{\frac{1}{0.8}} - e^{\frac{1}{(1000x+0.8)}}\right) - 2.4825\right)}{2.5}, \qquad Y_{rSOC} = [-1, 0] \tag{5}$$

$$U_{rSOC} = Y_{rSOC} \cdot \beta_{rSOC} \tag{6}$$

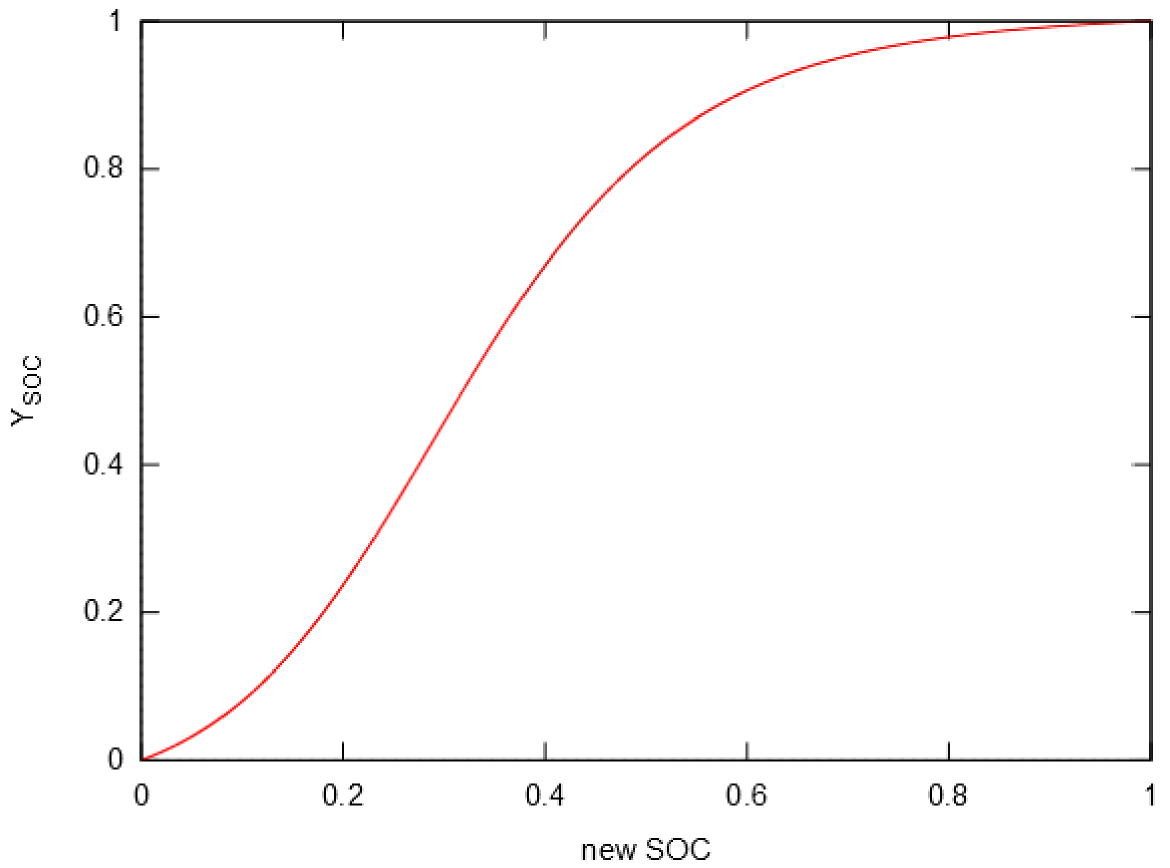
Y_{rSOC} reaches values between -1 for a serious need for charge and 0 for a sufficient SOC. To add it to the utility function Y_{rSOC} is weighted by β_{rSOC}.
The value of β_{rSOC} should be set to the amount of utility an agent would lose if he ran out of battery. This effects that a parking space which does not provide a fast enough charging infrastructure to reach enough state of charge for the agent to get to his next destination, will get a very low internal score.

2.3.2 Influence of the state of charge

Additionally to the aspect of having enough charge to reach the next destination, we assume that the state of charge influences an agent's well-being even if, from a rational point of view, he could fulfil his plan without charging.
Figure 3 shows the graph of equation 7, which describes the scaling function as it is assumed in the model. This definitely has to be subject of future work, since there is no empirical data supporting it yet.

$$Y_{SOC} = \left(\frac{\exp\left(-3.889x^2 + 10.97x - 2.159\right)}{2.425 + \exp\left(-3.889x^2 + 10.97x - 2.159\right)} - 0.0454411 \right) \cdot 1.067 \tag{7}$$

Figure 3: Scaling function for the new SOC, after the agent has parked on a specific space.



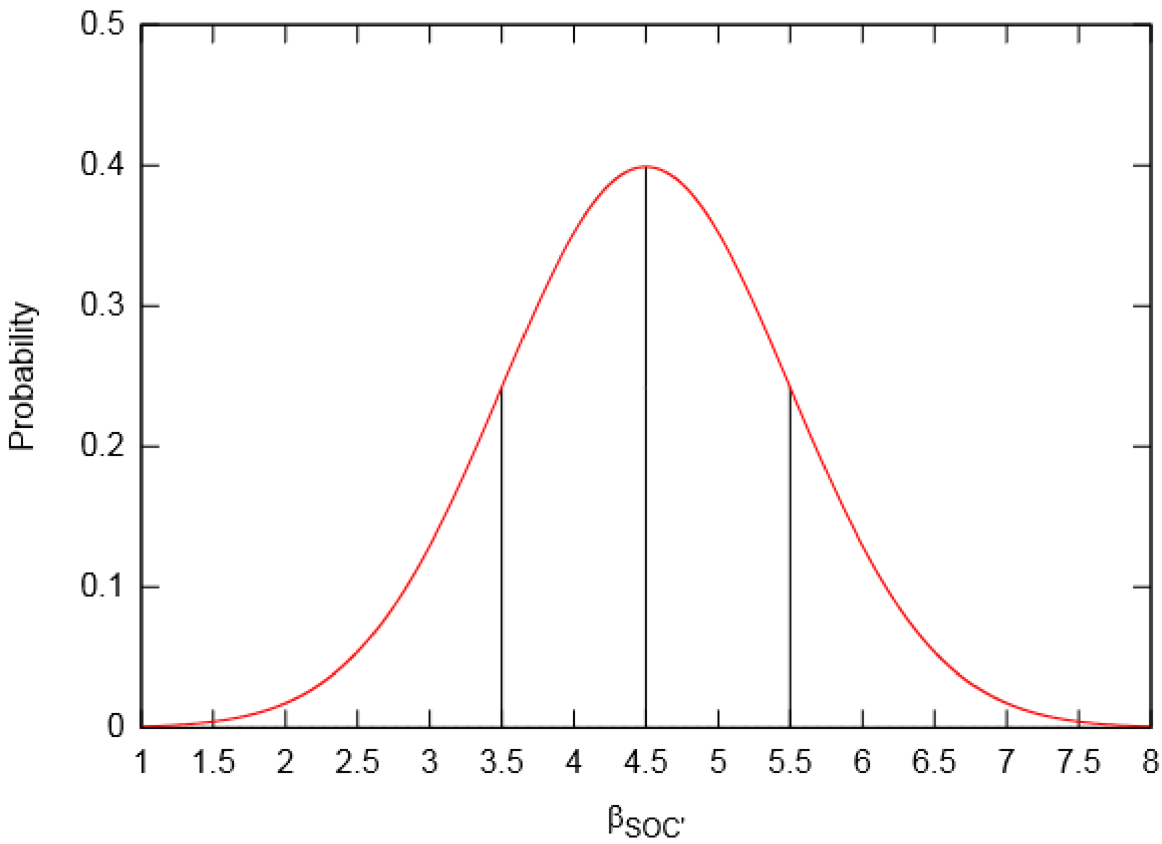
2.3.3 Consideration of varying charging behaviour

It is quite difficult to weight the value of a full state of charge without empirical data. Furthermore this kind of irrational behaviour probably varies a lot from person to person. Therefore we decided to normally distribute the β_{SOC} in equation 8.

$$U_{SOC} = \beta_{SOC} \cdot Y_{SOC}, \qquad U_{SOC} \geq 0 \tag{8}$$

$$\beta_{SOC} = \beta_{SOC'} \cdot \beta_{money}, \qquad \beta_{money} > 0 \tag{9}$$

Figure 4: Distribution of $\beta_{SOC'}$ (mean of 4.5$, standard deviation of 1$)



The midpoint of the distribution should be set to the value people are assumed to be willing to pay to have full charge. For $\beta_{SOC'}$ we used a mean of 4.5$ and a standard deviation of 1$ (figure 4). This means that 68% of the agents are willing to pay between 3.5$ and 5.5$ for 100%

SOC independently of the capability of reaching their next destination. These values seem to be quite high, but since EVs have less flexibility to quickly charge than conventional vehicles in case they would like to perform spontaneous activities, this appears reasonable. As described in equation 9 $\beta_{SOC'}$ has to be multiplied by β_{money}.

2.4 Examination of the model

As mentioned before there is no empirical data known to us to validate the model, but to give a better impression of the modelled behaviour we want to give an example. We start with an agent who has to chose between four available parking spaces. In table 1 the calculation is done for three different cases, in case 1 for a current SOC of 25%, in case 2 for 40% and in case 3 for 60%. It is assumed that, in all three cases, the agent needs 30% of his battery capacity for the rest of the day. This means, that in case 1 the agent is forced to charge his vehicle in order to reach his next destination, what is reflected in the resulting utilities for the different parking spaces. In case 2 the agent does not absolutely have to charge, but he takes the consequences of a worse conventional utility in order to gain a higher SOC, while in case 3 the agent is not willing to pay higher expenses, even if he would get a full charge.

3 Implementation of the parking choice model

The functional principle of the implementation is pretty similar to the one presented by Waraich and Axhausen (2012). But we decided to develop an entirely new implementation to gain more flexibility and to concentrate on the parking choice of electric vehicles from scratch. Nevertheless it is possible to use the implementation as a stand-alone unit without the support for EVs.

Table 1: Example for the decision between four parking spaces for three different states of charge (the required amount of energy for the rest of the day is 30% of the battery capacity)

case 1: 25% SOC				
Space	A	B	C	D
Cost	0	5	5	5
Distance	0	100	100	500
$U_{conventional}$	0.00	-0.33	-0.33	-0.43
new SOC	0.25	0.30	0.35	0.40
remaining SOC	-0.05	0.00	0.05	0.10
Y_{SOC}	0.34	0.46	0.57	0.67
Y_{rSOC}	-1.000	-0.993	-0.005	-0.001
U_{EV}	-29.90	-29.66	0.01	0.16
$\sum U$	-29.90	-30.00	-0.32	**-0.27**

case 2: 40% SOC				
Space	A	B	C	D
Cost	0	5	5	1
Distance	0	100	100	100
$U_{conventional}$	0.00	-0.33	-0.33	-0.09
new SOC	0.40	0.40	0.60	0.80
remaining SOC	0.10	0.10	0.30	0.50
Y_{SOC}	0.67	0.67	0.91	0.98
Y_{rSOC}	-0.001	-0.001	0.000	0.000
U_{EV}	0.16	0.16	0.25	0.27
$\sum U$	0.16	-0.17	-0.08	**0.19**

case 3: 60% SOC				
Space	A	B	C	D
Cost	0	5	5	5
Distance	0	100	100	100
$U_{conventional}$	0.00	-0.33	-0.33	-0.33
new SOC	0.60	0.60	0.80	1.00
remaining SOC	0.30	0.30	0.50	0.70
Y_{SOC}	0.91	0.91	0.98	1.00
Y_{rSOC}	0.000	0.000	0.000	0.000
U_{EV}	0.25	0.25	0.27	0.28
$\sum U$	**0.25**	-0.08	-0.06	-0.05

Figure 5: Example of how a parking lot is defined in the XML input file.

```xml
<Parking id="9000002">
   <capacityEV>200</capacityEV>
   <capacityNEV>800</capacityNEV>
   <chargingRate>8.04</chargingRate> <!-- in kW -->
   <parkingPriceMEVSpot>101</parkingPriceMEVSpot> <!-- pricing model
     for EV spaces -->
   <parkingPriceMNEVSpot>102</parkingPriceMNEVSpot> <!-- pricing model
     for conventional spaces -->
   <chargingPriceM>0</chargingPriceM> <!-- not used yet -->
   <facilityActType>parkingLot</facilityActType>
   <type>public</type>
   <coordinateX>684420.0</coordinateX>
   <coordinateY>4824402.0</coordinateY>
   <ocupancyStats>false</ocupancyStats> <!-- defines if stats for this
parking lot shell be written -->
</Parking>
```

3.1 Implementation of the parking infrastructure

Each parking space is represented as a single object. These spaces are pooled to parking lots. Each parking lot represents for example a companie's employee parking, the street parking in a street or an agent's private parking at home.
The parking lots are defined in an XML file where the following information is provided, separated into spaces with or without charging infrastructure for electric vehicles. An example for an entry in the XML input file is given in figure 5.

- id
- type (public/private)
- geographical position
- id of the facility the parking belongs to
- type of the facility the parking belongs to
- number of parking spaces
- rate of charge
- id of the used pricing model

3.2 Implementation of the pricing

Each parking can have up to two pricing models assigned to it, one for spaces with charging infrastructure and one for spaces without.

Figure 6: Example of how a pricing model is defined in the XML input file

```
<Parking_Pricing_Model> <!-- Secondary spaces without charging
  infrastructure-->
      <id>6</id>
      <maxTimeEV>24.0</maxTimeEV> <!-- not used yet -->
      <maxTimeNEV>24.0</maxTimeNEV> <!-- not used yet -->
      <priceOfFirstMinuteEV>0</priceOfFirstMinuteEV> <!-- for EVs -->
      <priceOfFirstMinuteNEV>0</priceOfFirstMinuteNEV> <!-- for
        conventional vehicles -->
      <pricePerHourEV>0.5</pricePerHourEV> <!-- for EVs -->
      <pricePerHourNEV>1</pricePerHourNEV> <!-- for conventional vehicles
        -->
      <evExclusive>false<evExclusive>
</Parking_Pricing_Model>
```

Standard, static pricing can be defined in an XML file as well, what makes changing policies easy even without any programming knowledge. An example for the definition of a pricing model in XML is given in figure 6. Each pricing model can be assigned to a varying number of parkings and should contain the following information:

- id
- maximum time of parking for EVs
- maximum time of parking for conventional vehicles
- price of first minute for EVs
- price of first minute for conventional vehicles
- price per hour for EVs
- price per hour for conventional vehicles
- EV exclusive (true/false)

Furthermore it is possible to implement dynamic pricing easily by extending the existing classes using Java. This makes it possible to vary the parking fees during the day, for example depending on the time or on congestion. An example for the integration of a dynamic pricing scheme is given in appendix A.6.
Because common systems like “park and charge” (Park & Charge, 2014) either provide flat rates or include the electricity rates into the parking prices there was no need for us to implement a possibility to define specific fees for charging yet. If this should be necessary it could still be integrated by implementing dynamic pricing as mentioned above.

3.3 The parking process

The parking choice simulation runs simultaneously with MATSim. The parking process is started by the time an agent reaches his destination in MATSim. Parking search related traffic is not considered.

3.3.1 Collection of available parking spaces

First of all the required information on parking lots within a specific radius of the agents destination is gathered. Therefore a collection of possible parking spaces is assembled. For conventional vehicles this collection contains one space for each available parking lot where a space without charging infrastructure is chosen if available.
For EVs the collection can contain one space with and one space without charging infrastructure for the same parking lot.

3.3.2 Information on the rest of the agent's plan

In the next step the selected plan of the agent is analysed to get information on the estimated parking duration and the estimated distance the agent will drive during the rest of the day, which is particularly important for the parking choice of electric vehicles. The estimated end time of the last parking period of the day is set to the activity end time of the first activity in the morning.

3.3.3 Parking choice

Using this information an internal score for each available parking space is calculated as described in section 2.
The required β_{soc} is calculated only once for each agent, so that an agent has constant behaviour during the whole simulation.
The parameters in the benefit functions and distributions described in section 2 are not hard coded so that they can be easily adapted. Furthermore the actual parking choice algorithm, containing the internal utility function, is placed in a separate class which could be replaced or overwritten without applying any changes to the rest of the implementation.

3.3.4 Feedback to MATSim

As described in section 1.1.1 the calculated score for each plan at the end of an iteration is crucial to the functionality of MATSim. Therefore the parking choice has to affect the MATSim scoring function to influence the agents behaviour.
For that purpose a parking related score is added to the MATSim scoring function. This parking related score contains the time the agent has to walk, the money the agent has to pay for the parking and for electric vehicles the money they save by charging their vehicle out instead of charging it at home (this value is negligible in most cases).

$$U = \beta_{walk} \cdot \frac{\text{distance}}{\text{walkingSpeed}} + \beta_{\text{money}} \cdot (\text{parkingFee} + \text{chargedAmountOfEnergy} \cdot \text{electricityRate}) \tag{10}$$

To keep the simulation consistent, the β values of the MATSim scenario used are utilised.
Unlike Waraich and Axhausen (2012) no changes to the agents plans are performed to take the time for extra walking legs into account. The implementation only affects the score by adding the disutillity for the walking time. With that simplification the time the agent walks is not subtracted from the time the agent performs other activities. This could be enhanced in future work.

3.4 Outputs

To be able to analyse the results some additional outputs are generated during run-time:

- The park history is an XML file containing information on all parking related events in chronological order. It gives information about every parking process including the agents id, the chosen spot, the state of charge and the internal score the spot reached. This file makes it possible to track the whole simulation what can be helpful for post processing and debugging. Additionally some EV related information is registered.
- The peak load file contains every parking, its position, its capacity, its type and its maximum occupancy rate during the iteration. This can be used for further analyses using GIS.
- The available parking chart gives information about how many parking spaces the agents have at their disposal depending on the time of day.
- Charts showing the walking distances to the chosen parking spaces.

- Charts showing the current occupancy rate during the whole day can be turned on for each parking.
- Some general statistics

Examples are given in section 6.2 and in the appendix.

4 Implementation of electric vehicles

Like for parking choice there are existing implementations for the integration of electric vehicles into MATSim as described in section 1.1.3. Even if the existing implementations provide way more detail, we decided to use a new one in this case to keep the simulation of EVs simple especially because this work does not focus on energy consumption. Another advantage of using a new implementation is the reduction of dependencies which simplifies the development. Nevertheless it is quite desirable to integrate an existing, more complex EV implementation into this parking choice package in a future work.

4.1 The electric vehicle

In this work only one type of electric vehicle is used, which is based on the "e-Up!" by Volkswagen (Volkswagen, 2013). This car was chosen because it can be found in the medium price segment and its battery characteristics are quite similar to the current models of other manufacturers as the BMW i3 and the Renault Zoe (cf. table 2).

Table 2: Comparison between three electric vehicles which are currently available on the market

	Battery Capacity [kWh]	Range [km]
Volkswagen e-up!	18.7	160
Renault Zoe	22.0	210
BMW i3	19.0	160

Source: (Volkswagen, 2013; Renault, 2013; BMW, 2013)

Every electric vehicle is assigned to a specific agent. Those assignments are defined in an XML file. Further the file contains the technical attributes of the EVs, what makes it possible to enhance the vehicle fleet.

4.2 Energy consumption

As described before the energy consumption of electric vehicles is not covered in detail in this work. Thats why an extremely simplified approach was used where the energy consumption is set to the average value of $11.7\frac{\text{kWh}}{\text{100km}}$ given by the manufacturer, without taking other parameters like velocity or road congestion into account. This leads to a range of 160 kilometres (Volkswagen, 2013).
During the simulation the energy consumption and the new state of charge of an electric vehicle are calculated every time it leaves a link. If an agent's SOC decreases below zero the agent gets penalised by a high disutility.
Furthermore agents driving an EV get additional utility each time they leave a link to compensate the lower energy expenses of electric vehicles compared to conventional vehicles. This is a workaround to be able to define two different values for the car cost per km which is used by MATSim.

4.3 Charging

Many manufacturers offer three types of charging infrastructure with different rates of charge. All three types provided by Volkswagen are implemented in the model.
To calculate the amount of energy, which is charged in a given time at given rate of charge provided by the charging infrastructure, a two-phase linear approach is used. The knee of the curve is located at 80% state of charge. To determine the two slopes of the curves the data from table 3 and the capacity of the battery is used to calculate an effective rate of charge out of the nominal rates.

Table 3: Charging devices provided by "Volkswagen" and corresponding charging times

	80% [h]	100% [h]
AC 2.3 kW	7	9
AC 3.6 kW	4	6
DC	0.5	

Source: (Volkswagen, 2013)

It is assumed that the agents leave home with a SOC of 80% but it is possible to define differing values in the XML input file for each agent. During the day an EV immediately starts charging at the time it is parked on a parking space which is equipped with charging infrastructure and stops charging if it either leaves the parking or reaches a 100% SOC. This charging behaviour is referred to as “dumb charging”.

5 The test scenario

5.1 The Sioux Falls scenario

The Sioux Falls scenario is a popular, small scale test scenario. It represents a simplified network of the city of Sioux Falls in South Dakota and is frequently used to test methods and algorithms for transport planning and simulation. The conventional Sioux Falls scenario is not suitable to be used in an agent-based simulation, because it is based on static origin-destination matrices (MATSim, 2014).

5.2 The Enriched Sioux Falls scenario

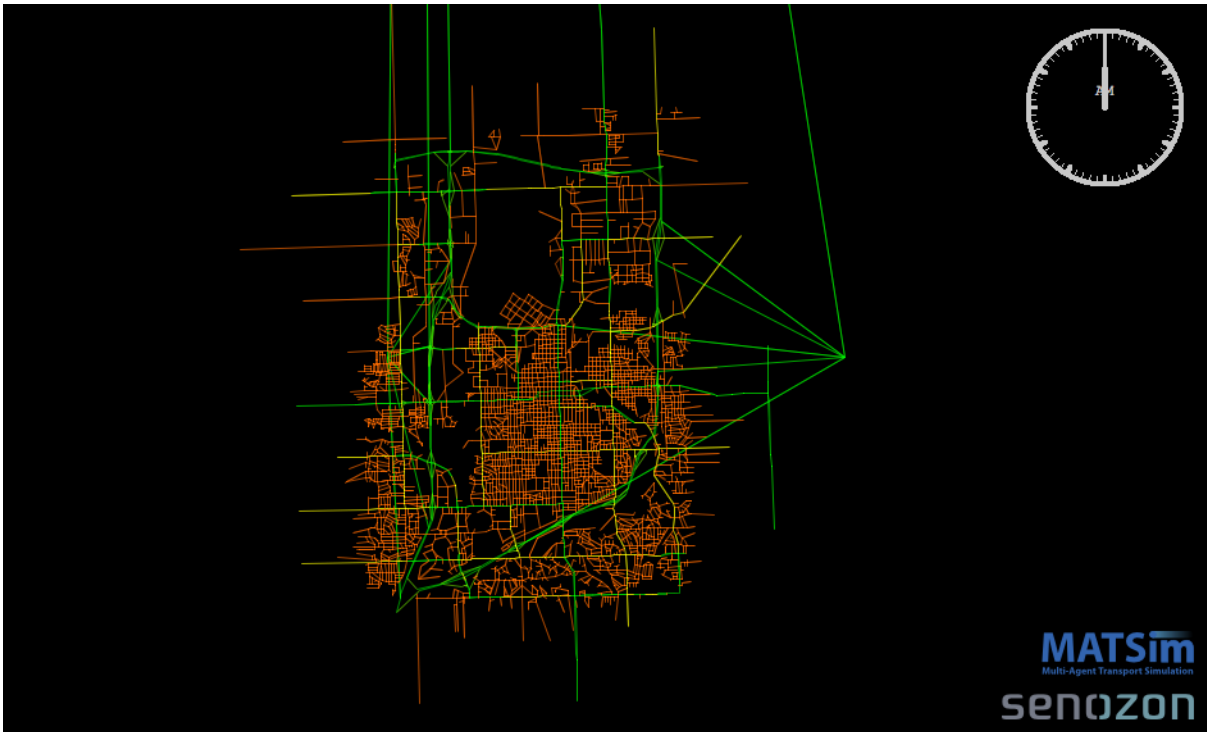
To use the Sioux Falls scenario in an agent-based simulation Chakirov and Fourie (2014) enriched it by dynamic demand. Based on real world surveys they created a synthetic population with socio-demographic agents as well as activity facilities with a high spatial resolution. They kept the network simple, even though they divided the links into shorter ones, connected by additional nodes due to the functionality of MATSim and because the agents start their route at the node closest to their facility and the distances between them would otherwise have been to big. To keep the scenario simple they restricted the plans to only two legs per plan, like home-work-home or home-secondary-home.

5.3 Scenario adaptation

5.3.1 Network

As the network of the Sioux Falls scenario is very coarse, it is not possible to create enough street parking and would also lead to large walking distances for agents, whose destination

Figure 7: Adapted Sioux Falls network with the links connecting the agglomeration in the north of the city. The agglomeration is not displayed.



facilities are located far off the grid. To solve this problem, a much more detailed network, based on Open Street Map data, is used instead. Thus the agents get the possibility to use a public parking close to their facilities. The detailed network causes much higher computational cost but a high resolution network is crucial for this kind of application.

An additional problem is, that the agents do not travel large distances (approximately 4 to 12 kilometre per day). As a consequence of this, the agents who use an electrical vehicle, only spend 2 to 4% of their battery capacities to reach their daily activity facility and have no obvious reason to charge their vehicle during the day. To create sufficient charging demand, we built in an agglomeration about 70 kilometre north of the city. The agglomeration is connected to the city with four major links, which split into several smaller links, to distribute the agents into the city and avoid congestions at the connections. The capacities of those links are set to high values, since they are not subject to the investigation. The adapted network is illustrated in figure 7.

5.3.2 Population

The population had to be adapted to the new network with the agglomeration. Therefore 30% of the population were randomly selected and moved to the agglomeration. To do that their facilities were duplicated to the new location, which for all of them is one single spot. In all cases the whole household was moved to the new facility, while the old facility persisted, in case it is used for another activity as well, like work or secondary. Furthermore the departure times of all moved agents were set 40 minutes earlier, to account for the longer estimated travel time. As this work is focused on the parking behaviour we decided to use only that part of the population which has a car available, so the population is now 65'758 out of 107'486 agents.

5.3.3 Parking

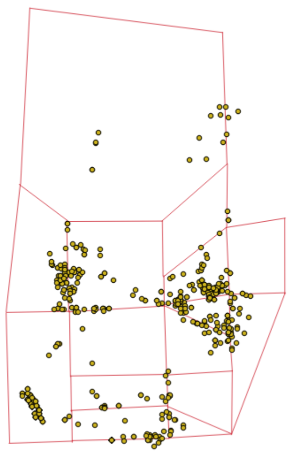
Because the scenario provides neither information about parking, nor is there sufficient data from other sources available, we had to find another way to integrate parking into the simulation. As mentioned in section 2.1 there are private and public parking possibilities. Public parking can be divided into two different types:

- street parking: the street parking capacity is depending on the length of the corresponding link. 100% street parking means one parking space every 5 metre.
- parking lot: parking lots have a high capacity of parking spaces. They are placed in areas of high parking demand.

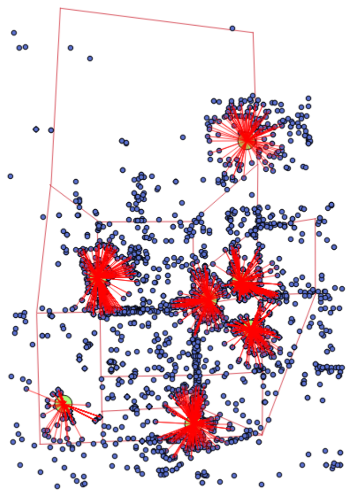
We assume street parking at every link and private parking at every activity facility. To determine the parking capacities, in a first step a scenario without any public parking and no limitations on private parking was simulated. Afterwards the capacities of all private parking, except for those corresponding to home facilities, were set to 80% of the received maximum occupancy. As it is assumed, that nearly everybody using a car has a private parking space at his home facility, the capacity of those were set to 100% of the maximum occupancy. Agents who did not use their car in the last iteration of the simulation therefore did not get a parking space at home. In a next step a scenario with the obtained capacities for private parking and additionally 5% street parking (one space per 100 metre) was simulated. Figure 8(a) shows the destinations of all the 600 agents who did not get a parking space within 1000 metre in that simulation. After that 7 parking lots, each with a capacity of 1000 parking spaces, were integrated, located in areas with high parking demand, as shown in 8(b). Finally the amount of street parking was raised from 5 to 10%.
With that setup we found a good way for every agent to have the possibility to get a parking but still to have pressure to find a parking space that is not too far from his destination facility.

Figure 8: Integration of the parking lots

(a) Destinations of all agents who did not get a parking space within 1000m in the simulation with only 80% private parkings and 5% street parking.



(b) Destinations of all agents who did not get a private parking space. The red lines illustrate destinations which are less then 1000m away from one of the 7 parking lots.



5.3.4 Electric vehicles and charging infrastructure

Electric vehicles were randomly assigned to 20% of the population. As described in section 1.1.3 there are other approaches, like the allocation based on the income, which could be used in the scenario setup for future work.

The three different charging devices, that are used in the simulation, with their charging times to 80/100% SOC are listed beneath:

- slow 7h/9h
- fast 4h/6h
- turbo 0.5h/-

They are distributed to the different kind of parking as follows:

- home 100% slow
- street 20% fast
- work 20% fast
- secondary 20% turbo
- parking lots 20% turbo

The slow charging device corresponds to an ordinary wall socket, what leads to the assumption, that slow charging is available at every parking space assigned to a home facility. Secondary facilities are assumed to provide turbo charging because the average duration of stay at those facilities is too short to gain sufficient SOC with slower devices.

5.3.5 Further adjustments

In the scenario proposed by Chakirov and Fourie (2014) a general disutility of $\beta_{0,car} = \beta_{tr,walk} \cdot 10\text{min} + \beta_{money} \cdot 6\$$[1] is used to consider parking. That value is not needed in this case and therefore set to 0.
Further the original enriched Sioux Falls scenario uses a car cost per km of 0.4\$[1]. Touring Club Suisse TCS (2014) states a value of 0.74 CHF which is approximately 0.96\$[1] for the overall car cost per km and an amount of 15% for fuel which leads to fuel expenses of 0.145\$[1] per km.
As mentioned in section 4.2 the average energy consumption of the used electric vehicles is assumed to be 11.7 kWh per 100 km. Using a electricity rate of 0.12\$[1]/kWh (Shrink That Footprint Ltd, 2013) this leads to energy costs of 0.014\$[1] per km for electric vehicles. Therefore the value described in section 4.2, which agents driving an EV get re-credit, is set to the difference of 0.131\$[1].
The walking speed is set to 4.86km/h (Weidmann, 2012) and the electricity rate is set to 0.12\$[1]/kWh (Shrink That Footprint Ltd, 2013).

[1] \$ = Australian Dollar

6 Experiments

6.1 Policies

The adapted Sioux Falls scenario was simulated using the following four parking pricing models:

- **A:** Free parking
- **B:** Normal pricing
- **C:** First minute 5$[1] for spaces equipped with charging infrastructure
- **D:** EV exclusive parking for spaces equipped with charging infrastructure

The pricing model used in case **B** is shown in table 4. That case is meant to be the initial situation for all other cases, except for **A**, which is just for testing purpose and does not consider any parking fees.

Table 4: Pricing model used in the case "normal"(**B**), prices in AUD per hour:

	charging infrastructure		no charging infrastructure	
	EV	conventional vehicle	EV	conventional vehicle
Work	0.5	0.5	0.5	0.5
Secondary	1.0	1.0	1.0	1.0
Street	2.0	2.0	2.0	2.0
Parking lot	5.0	5.0	5.0	5.0

Case **C** extends case **B** by an additional fee of 5$[1] for everybody who parks on a parking space that is equipped with charging infrastructure (EVPS[2]). This fee only has to be payed once and is not dependent on the parking duration. The intention of this extra fee is to keep vehicles without any need to charge away from EVPSs.

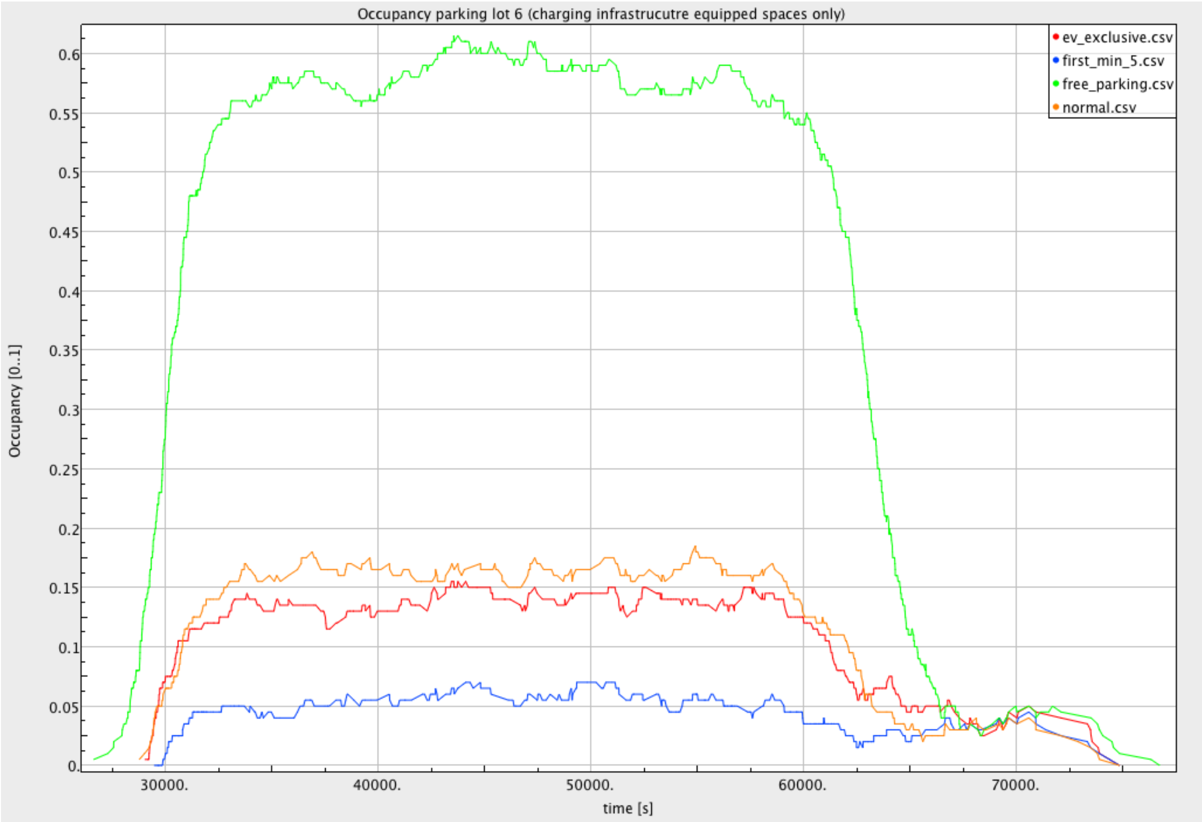
Case **D** uses the pricing model of case **B** as well, but extends it by the restriction that for conventional vehicles (CVs) it is only permitted to park on spaces without charging infrastructure (CPSs [3]).

For each case 200 iterations were performed, from which the last 50 were performed with re-planning disabled.

[2]Electric vehicle parking space (EVPS) refers to parking spaces equipped with electric vehicle charging infrastructure

[3]Conventional parking space (CPS) refers to parking spaces without electric vehicle charging infrastructure

Figure 9: Occupancy of the 200 spaces on "parking lot 6" which are equipped with charging infrastructure for the different cases.



6.2 Results

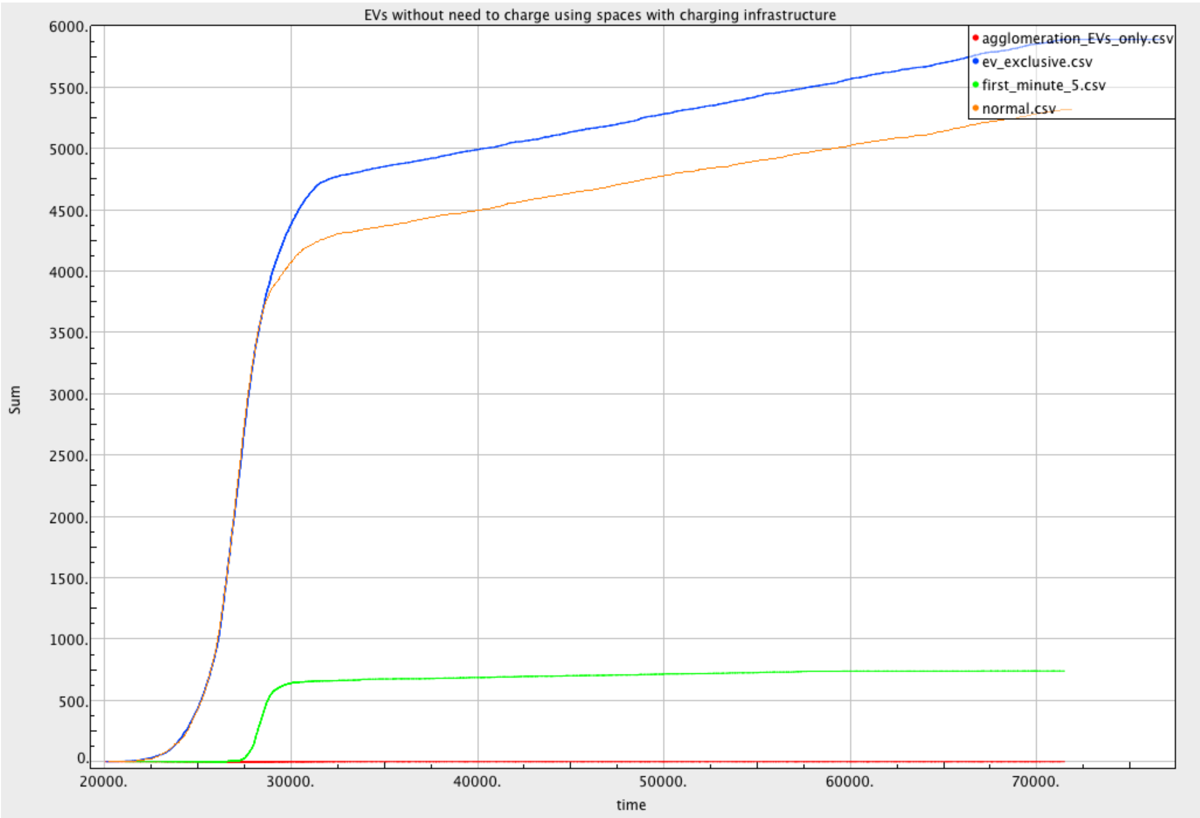
6.2.1 Parking lot occupancy

Figure 9 shows the occupancy of "parking lot 6", which is located in an area with a high density of work and secondary facilities, regarding only spaces which provide charging infrastructure. Since parking lots are the most expensive kind of parking, the agents try to avoid using them if they have other opportunities. Therefore the occupancy of the parking lots is a good indicator for the quality of the parking spaces the agents have at choice.

For case **C** a lower occupancy than for case **B** is observed, on the one hand this is caused by the higher availability of EVPSs at cheaper conditions and on the other hand by the fact that EVs using the parking lot without any need to charge decide to use a CPS.

Due to the missing difference between the prices of the parking lot and other parking opportunities the occupancy observed for case **A** is significantly higher than in all other cases.

Figure 10: The amount of EVs which do not have to charge but use parking spaces with charging infrastructure.



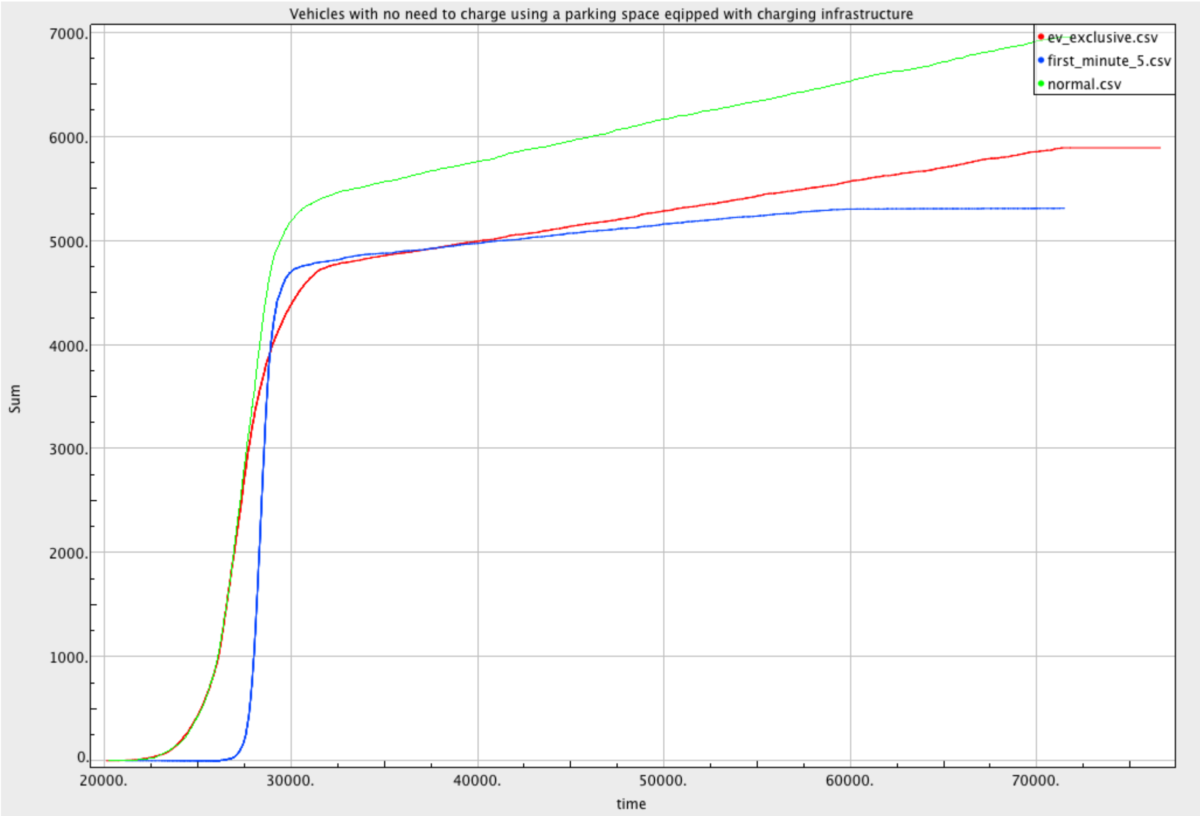
6.2.2 Efficiency of charging infrastructure equipped parking space usage

The possibility for EVs, which have to charge, to find a free EVPS depends on how many of the other agents use those spaces. Therefore we call the usage of EVPSs by agents who have to charge, to reach their next destination, efficient, while the usage by other agents is called inefficient.

Figure 10 illustrates that for case **D** more EV driving agents use an EVPS even if they do not have to charge. This behaviour can be explained by the higher occupancy of CPSs and the generally higher availability of EVPSs.

Furthermore the figure shows that for case **C** an inefficient usage of spaces with charging infrastructure by electric vehicles can be avoided. The sudden increase of the curve can be referred to the decreasing availability of CPSs, which forces the agents to take EVPSs. Figure 11 consideres both EVs and CVs. The curves for cases **B** and **D** run similar at the beginning, caused by the algorithm of the parking choice, which makes CVs choose CPSs as long as those are available. Therefore the behaviour of the CVs does not differ until there are no more available CPSs at their destination facilities. For case **B** the agents start using the EVPSs at their

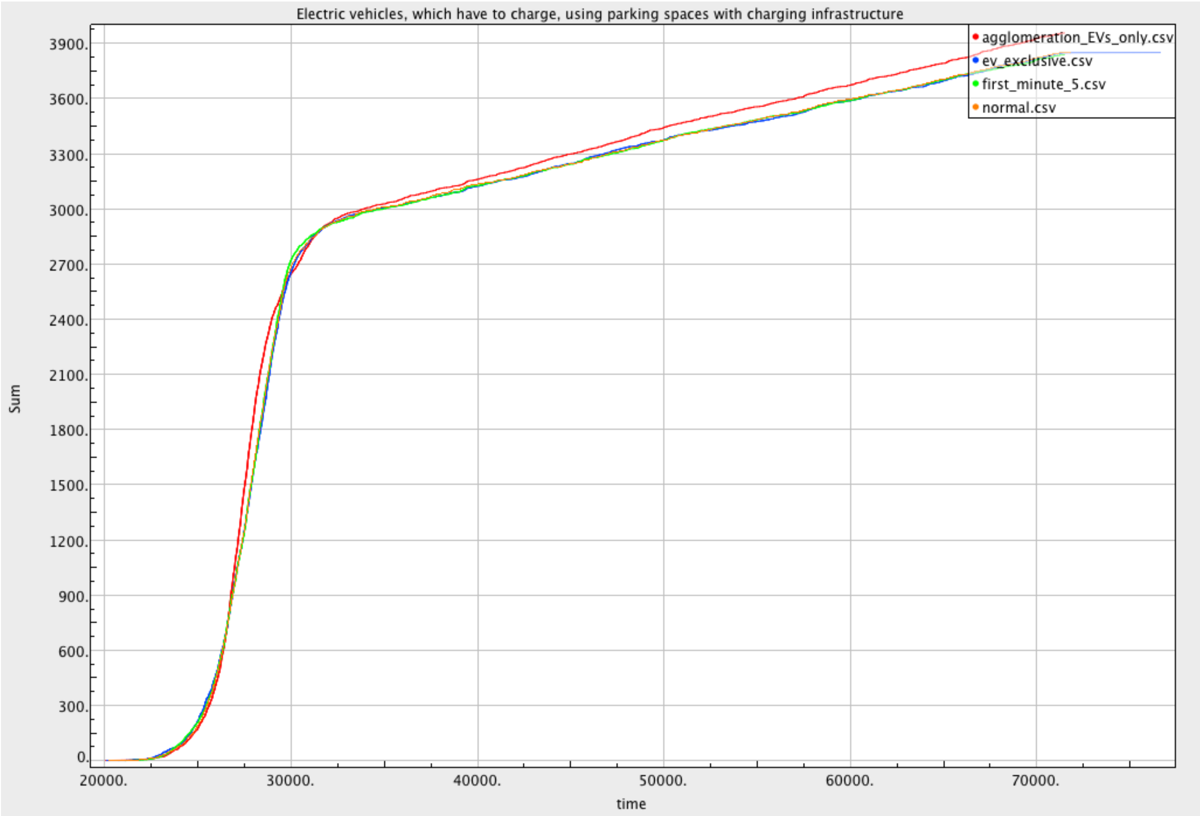
Figure 11: The amount of Vehicles (EVs and CVs) which do not have to charge but use parking spaces with charging infrastructure.



destination facilities at that point. For case **D** the agents driving CVs have to take public parking what is attended with higher expenses and walking distances.
The amount of CVs using EVPSs is much higher for case **C** than for any other case. This is due to the fact that all the EVs, which do not have to charge, try to use CPSs to avoid the extra fee of 5$[1], what leads to a higher occupancy of CPSs. Nevertheless case **C** accomplishes the lowest rate of inefficient usage.

The amount of EVs which have to charge and use EVPSs is shown in figure 12. The curves for all cases run quite similar. In order to find out the maximum efficient usage for the scenario, which is reached if a maximum amount of EVs that have to charge get an EVPS, a scenario containing only the EVs that have to charge was simulated. That scenario, which has no possibilities for inefficient usage of EVPSs is represented by the red line in figure 12. The shift of the increase to the left side is due to the lower traffic and therefore shorter travel times.

Figure 12: The amount of EVs, which have to charge and use parking spaces equipped with charging infrastructure.



6.2.3 Walking distances

To make driving an electric vehicle more attractive, agents driving an EV should not be disadvantaged by higher walking distances compared to CV driving agents. Figure 13 shows that the walking distances for EV driving agents could be reduced by the proposed policies. For case **C** as well as for case **D** it was possible to increase the number of EV agents parking at their destination facilities.

The distribution of the walking distances of agents who did not park at their destination facilities is shown in figure 14. The smaller walking distances for case **A** are due to the fact that the agents are not willing to walk to a distant available street parking instead of taking a parking space on a closer located parking lot since there is no difference in the pricing for that case.

The figure shows that especially for case **C** the difference between the walking distances of EV and CV agents could be eliminated and at least reduced for case **D**. Furthermore the spread of the walking distances of EV driving agents was significantly reduced for those two cases.

Figure 13: Histogram of the distances agents driving an EV have to walk from the chosen parking spaces to their destination facilities. The first bin contains all the agents which use the private parking at their destination facilities.

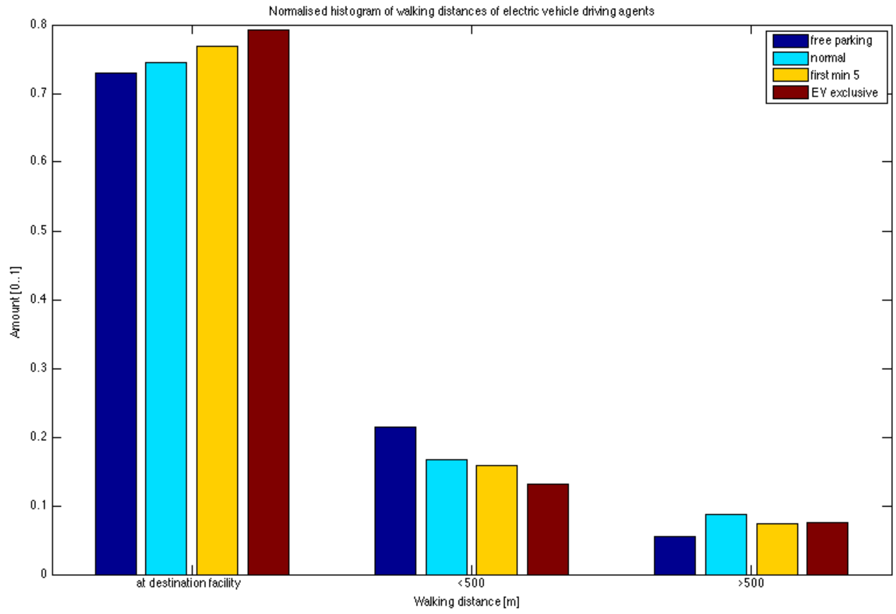
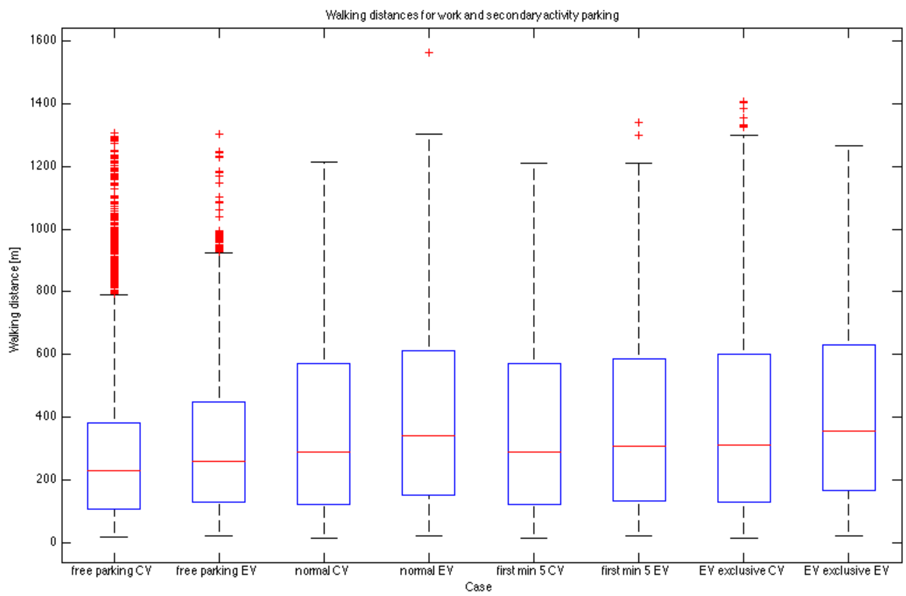


Figure 14: Distances the agents have to walk from the chosen parking spaces to their destination facilities. Agents using the private parking assigned to their destination facilities are not shown. (Maximum whisker length = 1.5 interquartile range)



6.2.4 Electric vehicles running out of battery

Of course one of the aims of all policies which should enhance the situation for electric vehicles is to avoid that EVs run out of battery. The number of EVs running out of battery for the simulated cases is shown in table 5. It turned out that those values vary a lot between different runs for the same case. That might be due to the quite small number of performed iterations.

Table 5: Number of electric vehicles running out of battery for each case averaged over two runs.

A: free parking	183
B: normal	157
C: first minute 5$[1]	126
D: EV exclusive	147
Agglomeration EVs only	117

6.3 Discussion of the policy studies

It could be shown that inefficient usage of EVPSs can be reduced by the proposed policies. However no significant increase of efficient usage of EVPSs could be observed. This means that the usage of EVPSs was only optimised in areas with uncritical demand for EVPSs compared to the amount of provided infrastructure. The lack of improvement in the areas of critical demand can be referred to the fact that case **B** is already close to the optimum. That is probably due to the circumstance that the situation in those areas is already optimised by the coevolutionary algorithm of MATSim itself. Agents driving an EV shift their departure times so that they distribute the demand for EVPSs over the day. Even though the amount of efficient EVPS usage could not be increased, the parking lot occupancy shows that, at least for case **D**, agents driving an EV had more convenient spaces at choice than for the initial situation.This is additionally established by the reduced walking distances.

Perhaps running out of battery could only be reduced but not be avoided due to a weakness of the adapted Sioux Falls scenario. The scenario in general has very few parking infrastructures, especially public parking, compared to the demand. Since the agents are only allowed to use either the parking assigned to their destination facilities or street parking or, if not too far away, one of the parking lots, there are only few options for the agents to change their behaviour.

Especially agents living in the agglomeration and performing secondary activities are reliant on parking spaces providing turbo charging infrastructure. Since those are only installed at secondary facilities and parking lots, agents performing a secondary activity more than 1000m away from any parking lot, cannot reach enough state of charge, if there is no EVPS available at their destination facility. Those limitations of the scenario are indicated by the results for the "agglomeration EVs only" simulation, which show that it is not possible to prevent all EVs from running out of battery. The results for case **C** were the closest to that minimum amount of EVs running out of battery. Even though the policy has two weaknesses, on the one hand the 5$[1] penalise all the agents driving an EV and living in the agglomeration, on the other hand 5$[1] are still not enough to force all the other agents, especially those who perform work activities, to use public CPS instead of using the EVPSs at their destination facilities if there is no CPS available at the facility.

7 Discussion and future work

Even though in the scenario at hand running out of battery could not be avoided, a general enhancement of the situation of electric vehicles by changing policies was shown. Therefore it seems to be possible to accomplish serious improvements in scenarios with less parking pressure and eventually by using more complex policies. The need for further investigations to this subject of study reasons the improvement of the parking choice model for electric vehicles and the methods which are required to test it in a future work.
Regarding the parking choice model there is a serious need for validation and calibration by empirical data. However doing a survey to gather this data is probably difficult due to the fact that there are only few people driving electric vehicles at present.
Additionally the capability to take all of the parking periods on an agents plan into account has to be implemented since this is crucial to be able to use the model to simulate scenarios containing more complex plans properly.

In this work it is assumed that the agents take off with a SOC of 80% in the morning. Therefore there is no need to charge for all of the agents who live in the city, which, combined with the assumption that everybody driving an EV has an EVPS available at home, seems reasonable. Nevertheless there might be cases in other scenarios where people do not have the possibility to charge at home or do not want to charge at home for example to save electricity costs. In those cases it would be necessary to either simulate several days per iteration or statistically distribute the SOC for those agents, which in terms of proper re-planing might get difficult.

Furthermore the evaluation of the simulation results showed that changing policies and regarding electric vehicles affects the simulation in a quite complex way, what makes it difficult to validate such a model. Therefore it would be useful to set up a scenario in a much smaller scale which gives the possibility to track every step of the simulation with reasonable efforts. Nevertheless the adapted Sioux Falls scenario proposed in this work should be improved. For one thing public transport, which is already used by Chakirov and Fourie (2014), should be integrated into the used, more detailed network. For another thing it would be helpful to divide the scenario into areas with different parking pressure to be able to test differentiated policies and gather results for areas of more or less critical demand.

8 References

BMW (2013) BMW i3 Technische Daten, webpage, www.bmw.de. Last checked: 13.05.2014.

Chakirov, A. and P. J. Fourie (2014) Enriched Sioux Falls Scenario with Dynamic and Disaggregate Demand, *Working Paper*, Future Cities Laboratory, Singapore-ETH Centre (SEC), Singapore.

Galus, M. D. (2012) Agent-based modeling and simulation of large scale electric mobility in power systems, Ph.D. Thesis, RWTH Aachen.

Galus, M. D., G. Georges and R. A. Waraich (2012a) Abschlussbericht des Projekts ARTEMIS (Abating Road Emissions Through Efficient (electric) Mobility - Interactions with the electric System), *Final Report*, Elektrizitätswerk der Stadt Zürich (EWZ), IVT, ETH Zurich, Zurich.

Galus, M. D., R. A. Waraich, F. Noembrini, K. Steurs, G. Georges, K. Boulouchos, K. W. Axhausen and G. Andersson (2012b) Integrating power systems, transport systems and vehicle technology for electric mobility impact analysis and efficient control, *IEEE Transactions on Smart Grid*, **3** (2) 934–949.

MATSim (2014) Multi-Agent Transportation Simulation, webpage, http://www.matsim.org. Last checked: 17.05.2014.

Park & Charge (2014) The charging infrastructure for electric vehicles, webpage, www.park-charge.ch. Last checked: 10.05.2014.

Renault (2013) Renault-Zoe-ZE-CH-DE-Broschuere, http://www.renault-suisse.com.

Rieser, M., C. Dobler, T. Dubernet, D. Grether, A. Horni, G. Lämmel, R. Waraich, M. Zilske, K. W. Axhausen and K. Nagel (2014) MATSim User Guide Development Version, 1 2014, www.matsim.org.

Schieffer, S. V. (2010) To charge or not to charge? decentralized charging decisions for the smart grid, ETH Zurich Institute for Transport Planning.

Shrink That Footprint Ltd (2013) Average electricity prices around the world, webpage, http://shrinkthatfootprint.com. Last checked: 19.05.2014.

Touring Club Suisse TCS (2014) Kosten eines Musterautos, webpage, http://www.tcs.ch. Last checked: 15.05.2014.

Volkswagen (2013) e-up! - Technik und Preise Gültig für das Modelljahr 2014, 12 2013, www.volkswagen.de.

Waraich, R. A. and K. W. Axhausen (2012) An agent-based parking choice model, *Arbeitsberichte Verkehrs- und Raumplanung*, **696**, IVT, ETH Zurich, Zurich, August 2012.

Waraich, R. A., C. Dobler, C. Weis and K. W. Axhausen (2012) Optimizing parking prices using an agent based approach, *Arbeitsberichte Verkehrs- und Raumplanung*, **794**, IVT, ETH Zurich, Zurich.

Weidmann, U. (2012) *Bahninfrastrukturen Kapitel 3: Infrastrukturprojektierung*, Lehrstuhl für Verkehrssysteme ETH Zürich.

A Documentation

A.1 Introduction

We want to give only a brief introduction how the implemented packages work and how they are integrated into MATSim. But please note that the whole project is still in an experimental state. Figure 15 shows the integration of the package into the MATSim controller. The interaction with MATSim is based on controller listeners, and a scoring function factory. The packages will work without the scoring function factory as well but there will be no feedback to MATSim in this case.
The required input files are already described in section 3.

Figure 15: Example of how the implemented packages can be integrated into the MATSim controller. Note that the parameters in quotes are usually set in the config file instead of being hard coded.

```
//setup controler:
Controler controler = new Controler(config);

//setup parking
ParkControlerListener parklistener = new
ParkControlerListener();

parklistener.setParkHistoryOutputFileName("parkHistoryFileName");
parklistener.getParkHandler().getParkControl().startup(
"parking_filename", "pricing_filename", controler);
controler.addControlerListener(parklistener);

//setup EVs
EVControlerListener evControlerListener = new
EVControlerListener();
evControlerListener.getEvHandler().getEvControl().startUp(
"evFilename", controler);
controler.addControlerListener(evControlerListener);

//Pass the EVControl to the ParkControl

parklistener.getParkHandler().getParkControl().setEvControl(
evControlerListener.getEvHandler().getEvControl());

//Setup the Scoring to give feed-back to MATSim
PlanCalcScoreConfigGroup planCalcScoreConfigGroup =
controler .getConfig().planCalcScore();
ParkScoringFactory factory = new ParkScoringFactory(
planCalcScoreConfigGroup, controler.getNetwork());
controler.setScoringFunctionFactory(factory);
```

A.2 Parking choice package

This package implements the parking choice algorithm as it is described in this work. It works as stand alone without the electric vehicle support as well. In the following is a short description of all implemented classes.

Class	Description
AdvancedParkingChoice	Implements the actual parking choice model which is described in this work. It gets all the available options as input, assigns scores to them and returns the best one.
IterEndStats	Generates some statistics and charts at the end of each iteration. All the data used for the charts is saved in CSV files as well.
ParkControl	Manages the whole parking process of one agent at a time. One instance of this class is kept by the ParkHandler which starts the park() / leave().
ParkControlerListener	Sets up event handlers and does some resets before and after each iteration.
ParkHandler	Handles the AktivityStartEvent and starts the parking process. Same for the ActivityEndEvent. Additionally all the events of each agent are counted to get the activity which the agent currently performs.
ParkHistoryWriter	Writes a file with parking related events. The start() method should be called before each iteration, the end() method should be called at the end of each iteration to write the file.
Parking	This represents a public / private parking lot. Every parking space is an own parkingSpot object. createSpots() creates the specified number of EVPS/CPS with the specified attributes. checkForFreeSpot() checks if one of those spots is available (functions for EVPS/CPS are available). clearSpots() should be called after each iteration to make sure there are no occupied spots at the beginning of the next iteration.
ParkingMap	Keeps a list of all parking in the network. getPrivateParking() returns an available private parking space at a specific facility. getPublicParking() returns available public parking spaces in a specific area.
ParkingPricingModel	Implementation of the static pricing models which are given in the XML input file. Calculates the costs for a given vehicle type and a given duration.

ParkingSpot	Represents a single parking space.
ParkingWriter	Does some JAXB magic to write a parkingMap with all its parking lots into an XML file.
ParkScoring	Takes the score from the VMScoreKeeper and adds it to the MAT-Sim score.
ParkScoringFactory	Adds the ParkScoring function to the default scoring function.
PricingModels	Keeps all pricing model objects.
PricingModelsWriter	Writes Pricing Models into an XML file using JAXB.
VMScoreKeeper	Is added as an agents attribute to keep all the parking related score.

A.3 Electric vehicle package

This package only adds a very simplified electric vehicle functionality to MATSim and does not work without the parking choice package. Figure 16 shows an extract of the used EV XML file.

Figure 16: Definition of EVs in the EV XML file

```xml
<entry>
 <key>27603_2</key>
 <value>
   <evType>e-Up</evType>
   <batteryCapacity>18.7</batteryCapacity><!--kWh--!>
   <consumptionPerHundredKilometre>11.7</consumptionPerHundredKilometre>
   <!--kWh/100km--!>
   <id>5719</id>
   <OwnerId>27603_2</OwnerId>
 </value>
</entry>
```

Class	Description
EV	Each electric vehicle is represented by an EV object.
EVControl	Processes the link enter and link leave events and is the interface to the parking choice package.
EVControlerListener	Sets up the event handler and sets all SOCs at the beginning of each iteration.
EVHandler	Lnk leave event handler. The event is forwarded to the EVControl.
EVList	List of all electric vehicles in the scenario.
EVListWriter	Writes the EV list for the scenario setup.

A.4 Util package

Class	Description
CSVReader	Reads CSV files and returns a LinkedList<String[]>.
CSVWriter	Writes CSV files.
ReadParkhistory	Reads a parkhistory file for post processing.
RemoveDuplicate	Removes duplicates from a LinkedList.
VMBoxPlot	Creates box plots from multiple data series.
VMCharts	Keeps data for many different charts and creates them at the end of each iteration. New charts can be added easily.

A.5 Scenario adaptation package

The classes in this package are only for our special adaptation purposes. Some of them might be useful in other cases as well but in general those are meant to be examples.

Class	Description
AdjustParkingCapacities	Required inputs: parking XML file, CSV file containing peak occupancies, CSV file containing the maxmimum amount of parking spaces for each street parking. Output: parking XML file with adjusted capacities and adjusted charging rates.
AssignEVs	Randomly assigns EVs to a given amount of agents. Output: EV XML file.
CreateDemoParking	Input: scenario. Output: parking XML file containing a parking at each facility and each street.
CreateDemoPricingModels	Creates pricing models with given values. Applying changes directly to the XML files is probably more convenient.
NetToCSV	Input: XML network. Output: CSV (for GIS analyses).
RemoveAgents	Removes agents which do not have a car available from the population.
VMScenarioTool	Script to setup new scenarios.

A.6 Example for dynamic pricing

A short example for the implantation of a dynamic pricing model is given in figure 17. The example uses the current occupancy of the parking lot and separates between conventional and

Figure 17: Example for a very simple implementation of dynamic pricing. It separates between EVs and CVs and takes the current occupancy of the parking lot into account.

```
public class SpecialTestModel extends ParkingPricingModel {
  public int id = 0;

  public double calculateParkingPrice(double duration,
 boolean ev, ParkingSpot spot){
    duration = duration/3600; //seconds to hours
    double occupancyEV = spot.parking.getOccupancyEVSpots();
    double price = 0;
    System.out.println("test model in use");
    if (ev){

      price = 10; //for electric vehicles
    } else {

      price = 20; //for conventional vehicles
    }

    if(spot.isCharge()){//extra fee if a parking space
               //equipped with charging infrastructure is used
      price += 5*occupancyEV; //increases with the occupancy

    }
    return price
  }

}
```

electric vehicles and checks whether it is used for a parking space with or without charging infrastructure. There are many more influences which could be considered as for example the time.

Figure 18 shows how a static pricing model, which was defined in an XML input file could be replaced by the implemented dynamic example model. In this case every parking which pricing model 0 was assigned to would use the dynamic one instead. The shown code snipped can be placed in the controller right behind the setup of the parkControl.

Figure 18: Example of how a static pricing model could be replaced by a dynamic one

```
PricingModels pricing =
parklistener.getParkHandler().getParkControl().getPricing();
pricing.removeModel(0); //Remove the static model which
should be replaced
//Instantiate the dynamic model:
SpecialTestModel testModel = new SpecialTestModel();
pricing.add(testModel); //Add it to the PricingModels list
```

A.7 Example for post processing

Post processing is very useful to generate outputs which are not generated by default without running a new simulation. For example it can be interesting to track the behaviour of one single agent or to get a list of all agents which have a certain behaviour. Therefore we tried to record the maximum information in the parkhistory file. This information can be accessed by the ReadParkhistory class. An example is given in figure 19.

Figure 19: Example for a simple post processing task. The script prints and counts electric vehicles which use an EVPS and still run out of battery.

```
ReadParkhistory hist = new ReadParkhistory();
hist.readXML("normal_sa/parkhistory/parkhistory_200.xml");
int i =0;
HashMap<String, Boolean> outOfBattery = new HashMap<String,
Boolean>();
for(HashMap<String, String> event :
hist.getAllEventByAttribute("eventtype",
"ev_out_of_battery")){
  outOfBattery.put(event.get("person"), true);
}
for(HashMap<String, String> event :
hist.getAllEventByAttribute("eventtype", "EV_parked")){
  if(checkIfHome(event.get("person"), personList)){
    continue; //only consider parking during the day
  }
  if(outOfBattery.containsKey(event.get("person"))){
    if(event.get("spotType").equals("ev")){
      i++;
      System.out.println(event.toString());
    }
  }

}

System.out.println(i);
```

B Outputs

In this section some example outputs are given.

B.1 Parkhistory

Figure 20 shows an extract of the parkhistory file.

Figure 20: Extract of the parkhistory file


```
<parkevent time=28351.0 eventtype=NEV_parked person=40702.3
parking=42970 spot_score=-1.2990813320643462 parkingType=public
spotType=nev>
<parkevent time=28351.0 Parkingid=42970 Parkingtype=public
eventtype=occupied last_person=40702.3>
<parkevent time=28353.0
eventtype=Agent_looking_for_parking_has_to_charge person=36400.2>
<parkevent time=28353.0 eventtype=EV_parked person=36400.2 parking=39629
spot_score=-0.9027633828452124 parkingType=public spotType=ev
stateOfChargePercent=27.631231980037246>
<parkevent time=62149.0 eventtype=EV_left person=36400.2 parking=39629
parkingType=public spotType=ev stateOfChargePercent=100.0>
```


B.2 Charts

The figures 21-23 show some example output plots for the 200th iteration for case B. Every Chart is separated into EVs and CVs.

Figure 21: Amount of parking possibilities the agents have at choice when they reach their destination.

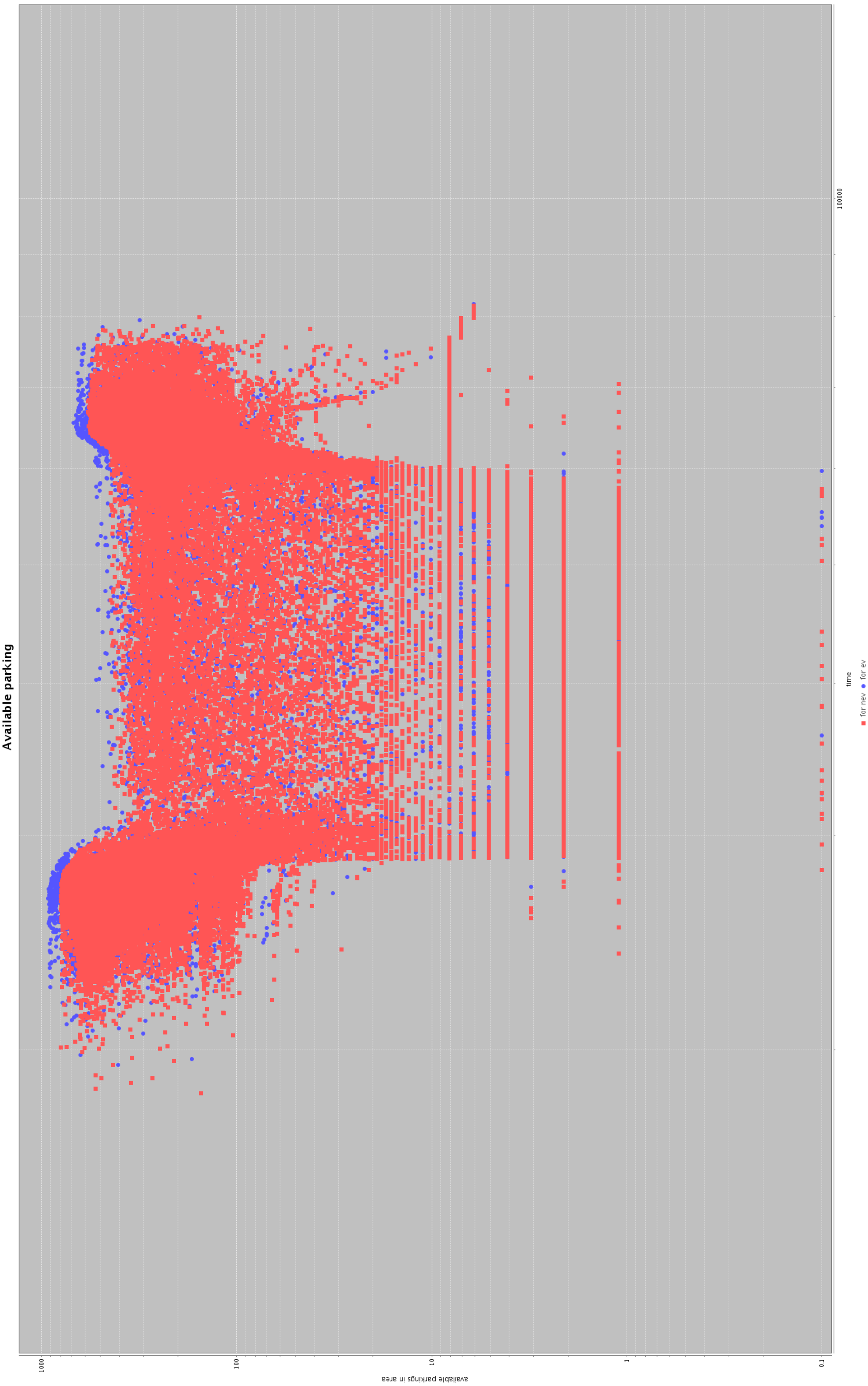


Figure 22: The distance the agents have to walk from their chosen parking space to their facility

Figure 23: The daily utility the agents receive versus the distance they travelled during the day