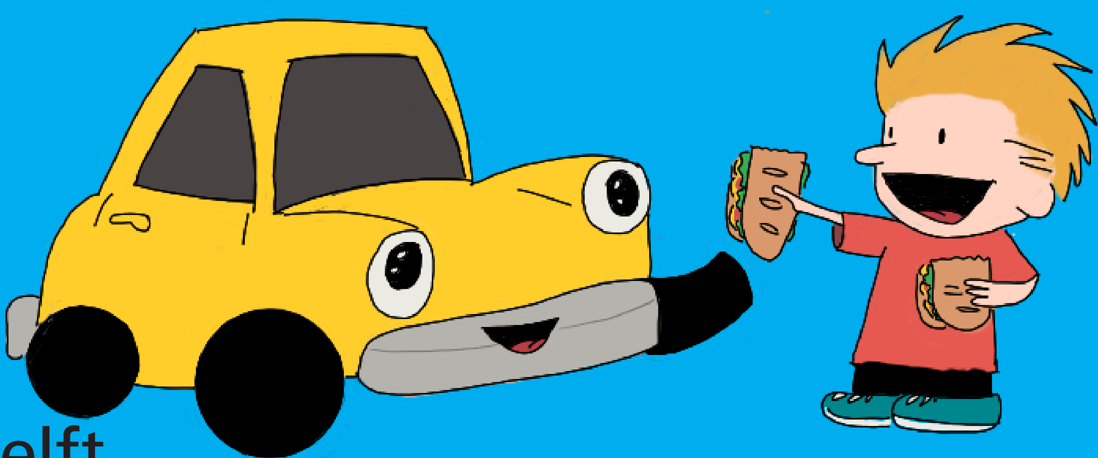
Dynamic planning for recharging shared electric taxies

H. Jamshidi



TUDelft

Dynamic planning for recharging shared electric taxies

by

H. Jamshidi

to obtain the degree of Master of Science
in Transport and Planning
at the Delft University of Technology,
to be defended publicly on Tuesday September 24, 2019 at 11:00 AM.

Student number:	4721837	
Project duration:	Feb, 2019 – Sep, 2019	
Thesis committee:	Prof. dr. ir. B. Van Arem,	TU Delft, committee chair
	Dr. ir. G. H. D. A. Correia,	TU Delft, daily supervisor
	Dr. ir. J. T. Van Essen,	TU Delft, external supervisor
	Dr. ir. K. Nökel,	PTV Group, company supervisor

Preface

This thesis marks the completion of the Master Civil Engineering, track Transport Planning at Delft University of Technology. I worked on this thesis in cooperation with the PTV Group. The aim of the thesis was to allow an electric taxi operator, better manage its charger infrastructure, and to allow it to make more profit. Hopefully, new generation of smart charging planners, can facilitate the transition towards an electric mobility. It was a long and winding road, from which I had to learn a lot to survive, from modeling techniques, to managing time and stress.

I would like to thank the people directly involved in this thesis. Many thanks to Klaus for the genuine curiosity in the work, also for making me feel important, doing this work. One comes in true need of that, in a long eight month. Thanks to everyone in PTV, especially Nitin, for hosting me in Karlsruhe, and patiently responding to my many emails. Thank you Goncalo for the support, for the meetings, and for generously offering to help me with my presenting skills. Thanks to Theresia for reading my intermediate reports, with a precision, that maybe even I did not have the patients for. I owe it to all the feedbacks, that the current content is decipherable to the next reader(s), if any. Thanks to my chair, Bart, for allowing flexibility on requirements of almost every meeting we had, and trusting that I will get there in time. Thanks to Natalia, for our two brain storm sessions, and whose insistence on starting with the smallest problem possible, lead me to the main algorithm in this thesis.

Thanks to all the people staying late at Afstudeerhok, and Fatemeh at Vasat, without company of whom, I should have extended this project by another month at least. Thank you Kian, for tolerating the worst of me during the last months. Not to leave out the artist of the great cover, Ghazaleh, and my fateful editor, Aida. Last, but definitely not least, I like to express the deepest gratitude to my mom, dad, and my sister Jana for being the kind individuals that they are, but also for almost everything else.

A final, yet necessary thanks to Russ Roberts, for his podcast, and many many interesting hours. Specially, for the reminder, that it works wonders for productivity of our jobs, if we have skin in the game. Particularly to be extra careful in academia, where us practitioners can give recommendations for a living, often while being completely unaffected by their consequences. That brings me to doing a thesis on the recharging problem, where of course I had the motivation to make a sophisticated sounding contribution. But next to it, I also wanted to answer, would I use the charging planner that I develop for my own taxi service, or the one that I have invested my own money in. The answer is not yet affirmative, but at least I graduated trying.

Helia Jamshidi
September 17, 2019

Executive summary

0.1. Introduction

The persistent growth of urban population poses environmental, economic and social challenges to transportation systems that move the people, goods, and services of society. Urban mobility has traditionally been restricted to privately owned vehicles or public transport. However, advancements in technology and ICT have introduced innovative dynamic transit services that are radically changing the traditional views of the transportation industry. Mobility as a Service (MaaS), combines different transport modes to offer a tailored mobility package, similar to a monthly mobile phone contract and includes other complementary services, such as trip planning, reservation, and payments, through a single interface [10].

Electrification of the mobility services offers additional environmental benefits from decreased urban emissions, to higher energy efficiency. Operational cost is also lower, because of cheaper energy, lower energy expenditure per kilometer (as they can restore charge while driving downhill), lower repair costs, and longer lifetime [2].

Despite the environmental benefits of a transition towards zero-emission vehicles, the diffusion of electric vehicles (EVs) faces barriers such as range limitations, limited charging stations, and long charging and queuing times. The driving range is even more restraining in hilly regions, in winter temperatures and when the driver wants to run heating and air conditioning. While private owners can rely on overnight charging, for their daily use, taxi drivers are not viable without at least charging once during the day.

A long term study on charging patterns among E-taxies (ET) in Shenzhen [20], which has the largest number of E-taxies in the world, reveals interesting findings on charging time and location preferences of drivers, and points out disadvantages such as capacity losses at charging stations, resulting from lack of full information, and sub-optimal planning by individual drivers. They reported that 20% of charging stations accommodate 80% of charging demand in 2013 due to their convenient locations, e.g., most of them are located in downtown areas, where the possibility for ETs to pickup passengers after charging is high. They concluded that charging stations play a key role in large-scale ET promotion and even though enough charging points are deployed, the unbalanced temporal or spatial charging demand and supply reduce the efficiency of the overall charging network. The experience in Shenzhen suggest that more efficient utilization of the charging infrastructure, and smooth operation of E-taxies, require a software backend that tracks vehicles, their State of Charge (SoC) and customer ride requests to manage and balance the electric fleets. The charging planner should navigate the driver to the best charging location based on the availability of chargers, remaining battery range, and overall demand for charging.

This study aims to extend the functionality of the MaaS operation software at PTV to include Electric Vehicle (EV) operations. PTV MaaS Modeller is a software component portfolio that supports city administrations and MaaS operators from planning through an operation to the management of their MaaS processes. *MaaS Dispatcher*, the module of interest in this study, targets the “pickup and delivery problem” where a fleet of vehicles operates on a road network in such a way that it can serve requests with temporal constraints. The specific use case is the operation (or simulation) of an on-demand ride-pooling service in a city or larger area.

0.2. Problem statement

We are going to study a system defined by an operator who owns a fleet of electric taxies and earns revenue from satisfying requests that come in real-time from passengers going from an origin to a destination. The operator has to assign a vehicle to each request within a short period. For this project, it is assumed that passengers request the service, knowing their ride might be shared, and subjected to detour. The operator is constrained by its resources and the demand. The size of the fleet and the number of available chargers are hard constraints in this system. Vehicles can only operate if they have enough charge. Any shortage in the number of charged cars or wrong distribution of charged cars within the operating zone will result in the rejection of potential customers. We cannot charge cars if we need them for operation, and we cannot use more chargers than we have. Therefore, the number of vehicles we can charge at any moment is constrained both by the number of vehicles and the number of chargers.

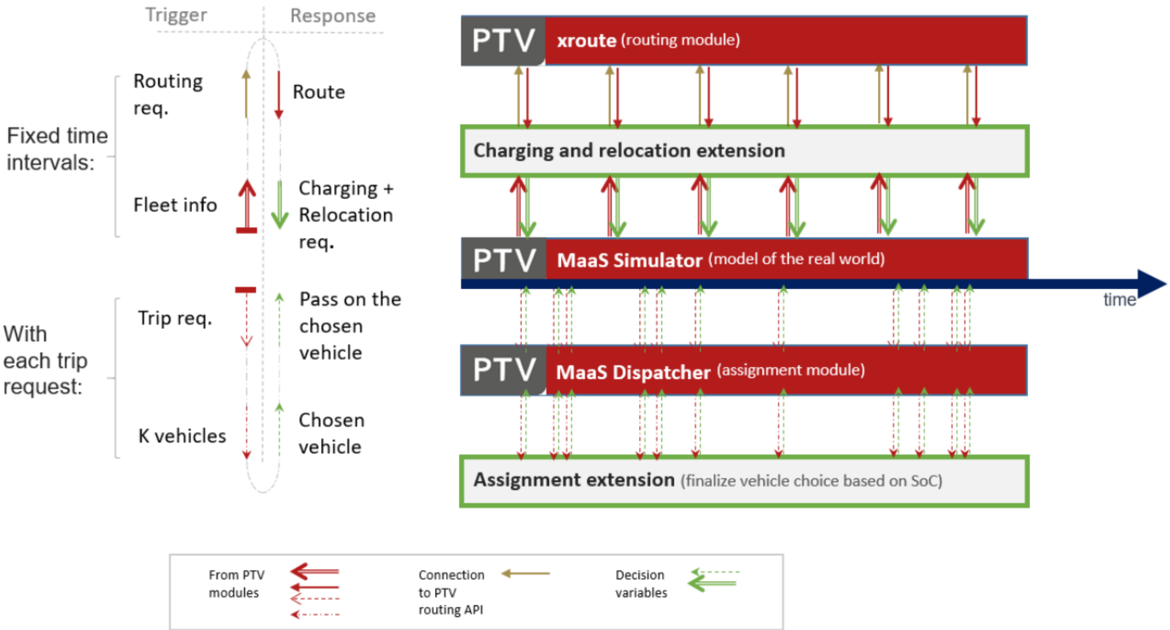


Figure 1: Connection with PTV modules

As mentioned, this project will accompany a series of services by PTV; namely *xroute* for The routing problem and *MaaS Dispatcher* for the assignment problem. The *MaaS Dispatcher* will assign vehicles to requests that are realized in real-time. The requests are processed one by one, in a first come first serve manner; a vehicle assigned to a new request can already have passengers on board; this means that *MaaS Dispatcher* is also deciding how passengers rides are shared. *xroute* is used for routing between known waypoints (pickup, drop-off points, or any specified coordinate), it is used by the *MaaS Dispatcher* and will also be used in the charging planner, to get information about the distance of the vehicles to the charges. In order not to interfere with the algorithms that already exist in PTV, ideally the charging problem should be solved independent of the routing and the assignment problem.

The existing PTV Dispatcher can assign vehicles to requests, under the condition that vehicles have an infinite battery. It has no means to monitor the SoC, to send vehicles to recharge and to make sure they do not run out of charge during operation. Therefore two additional mediums are required to ensure smooth operation of the fleet. One is sending vehicles to charge, with a specified duration. Another is to prevent the dispatcher from choosing vehicles for trips they do not have enough charge for.

Figure 1 illustrates the interactions between the existing PTV modules, and the algorithmic extensions that are the products of this thesis. The first means of communication is addressed with PTV Dispatcher defining a special kind of trip request with no origin, just a destination. Once requested, the trip will be enforced after the last planned drop-off of the vehicle. The second interaction point of the charging planner with the dispatcher is triggered after each trip request has been activated. It is to guarantee that a vehicle has enough charge to make it to the closest charging location after its last drop-off. After each request is submitted, the dispatcher will find k vehicles that can perform the trip, and return the k options to the *Assignment extension*. The *Assignment extension* can then choose the best one according to the vehicles' SoC.

The *MaaS Simulator* is used to evaluate the case study, designed to test the effectiveness of the algorithms developed. It mimics the real world, namely the road network and the passenger requests. Four modules are employed to operate the vehicles cost effectively; they are provoked by the simulator as needed. All component from PTV are red in color in Figure 1; likewise, red arrows mean that they are constructed by the PTV components. The remaining blocks with green outline are products to be designed in this study. With each request the *MaaS Dispatcher* is actuated, once the *MaaS Dispatcher* finds k best option for the request, the *Assignment extension* can finalize the option based on SoC of the k options. At fixed time intervals, simulator triggers the *Charging and relocation extension*, which has access to *xroute* through an API and can acquire as many routing requests as it needs, to decide if any vehicle should relocate to another zone or go to a charging station.

The operator only uses private charging stations, hence there is no competition with exogenous vehicles for the charging spots. The chargers offer either slow charging or fast charging. The electricity prices are variable throughout the day, but they are known beforehand. More restrictive scenario parameters include homogeneity of the fleet, in terms of fuel type, capacity, consumption rate, driving range, charging compatibility. The charging curve is a piecewise linear function with a lower charging rate after SoC of 80%. The capacity for all vehicles is six passengers. The size of the fleet is also fixed throughout the day. Finally, the consumption rate is assumed solely a function of distance for the case study.

Having shaped the scope, we can outline the abstract problem of dynamic recharging and relocating of electric fleets. The problem decisions, objectives and constraints are as follows:

Decisions:

1. The decision to mandate a specific vehicle to go to a charging station after finishing its current tour. These decisions can be made in fixed time intervals (e.g. every 4 minutes)
2. The decision to mandate a specific vehicle to relocate after finishing its current tour. These decisions can be made in fixed time intervals (e.g. every 4 minutes)
3. The vehicle to be chosen for trip requests among a restricted set, given by the *MaaS Dispatcher*

Objectives:

1. Level of empty routing (going to charging station, and re-positioning) is kept to a minimum
2. Low electricity cost
3. Serve as many passengers as possible

Constraints (including but not limited to):

1. No vehicle runs out of charge in the middle of the operation
2. Passengers waiting time is restricted
3. Fleet size is restricted
4. There are limited charging stations
5. Capacity of charging stations are restricted
6. Vehicle capacity is limited
7. Charging rate is restricted

After the algorithms for *Charging and relocation extension* and *Assignment extension* are presented, their performance is put to test in a case study in Barcelona, where the results from the developed algorithm are compared to a simple charging rule.

0.3. Method

0.3.1. Charging and relocation extension

Algorithmic outline

We need to decide in real-time when, and where each vehicle goes to charge, when each vehicle stops charging, alongside moving vehicles between the zones when necessary. However, we do not know the energy needs of individual vehicle for the rest of the day, or the next trips to be picked up by the vehicle. The information that the algorithm has access to includes the number of active vehicles required through the day (aggregated per time step), and reliable information on the number of pickups and drop-offs per zone per time step. There exists a gap between the information we have, which is aggregated both in space and time, and the decision we have to make that is to move a specific vehicle at a particular time to a specified charger or coordinate (for relocation). Three sequential algorithms are proposed in this study to connect the aggregated information we have to the decision we have to make in real-time for individual vehicles. The three algorithms are named A, B, and C. Sequential means that the output of the first algorithm is passed on as constraint to the second, and the output of second likewise passed to the third algorithm. The rest of this section is dedicated to explaining the main purpose of each algorithm, and the idea behind their modeling.

Algorithm A's purpose is to look at the overall expected energy consumption throughout the day and set the aggregate charging schedule. It will also make sure that vehicles are not sent to charge in demand peak hours if the operator needs them for serving passengers. Algorithm A is interpreted as operator's daily plan for charging.

Algorithm A answers the questions:

1. How many vehicles should go to charge in each time step of 30 minutes during the day?
2. How many vehicles should be in charge in each time step of 30 minutes during the day?

Algorithm B is designed based on the zones in the operation area. The zones come from the prediction data on the number of expected pickups and drop-offs, which we assumed the operator has. The algorithm

controls flow between the zones, and does not decide on individual vehicles. It makes sure that charging locations are close to the areas where we expect to find passengers.
Algorithm B answers the questions:

1. In the next 30 minutes how many vehicles should go from each zone to each charger?
2. In the next 30 minutes how many vehicles should go back from each charger to each zone?
3. In the next 30 minutes how many vehicles should move between the zones?

Algorithm C chooses the vehicles that fulfill the flows from algorithm B. It has complete information about the vehicles. It aims to minimize the cost of empty routing (relocation and charging trips).
Algorithm C answers the questions:

1. Which vehicle should go to each charger after its current tour?
(only if the vehicles' tour ends before the next call to *Charging and relocation extension*, here 4 minutes)
2. Which vehicle should go to each zone after its current tour?
(only if the vehicles' tour ends before the next call to *Charging and relocation extension*, here 4 minutes)
3. Which vehicle should stop charging now, and which zone should it move to?

Algorithm A

Algorithm A is a Mixed Integer Linear Program (MILP) that handles the charge dynamics of the fleet. By charge dynamic, it is meant that dimensions representing one vehicle are reduced to what takes place in its battery. It does not matter where the vehicles are, or which trip they are serving. In this system, a vehicle can only be active, idle, slow charging, or fast charging. To make the problem simpler, the operation day is divided into time steps of 30 minutes.

Demand is given as input to the algorithm, the more demand the fleet can satisfy the better. Vehicles pay for charging and are penalized every time they start to charge. This is to prevent a plan with frequent and short charging periods. The price of charging is per KW gained, and relative to the price of overnight charging in the depot. In the real scenario, the operator not only has to have enough charged cars but also have them scattered spatially; hence a parameter is introduced as the desired number of charged cars for each time step. Satisfying the number of desired charged cars is rewarded in the optimization model.

Algorithm B

Algorithm B is an aggregated level MILP optimization to take advantage of aggregated pickup and drop-off information over a two-hour horizon. The aggregated decisions are empty vehicle flows between zones over which data on pickup an drop-offs are available. Next to the operating zones, there are charger zones, each including a bundle of charging stations close together. The charger zones do not need to be within boundaries of normal zones. The output of aggregated level optimization (algorithm B) is then the number of vehicles moving between operating zone, going from operating zone to charger zones, and from charger zones to operating zones.

The first input of the algorithm is the initial location of the vehicles and their Expected Arrival Time (EAT). Algorithm B also gets the number of vehicles that have to be in charge in each time step, along with the least number of vehicles who have to go to charge from algorithm A. The second is required to distribute charge among vehicles, otherwise algorithm B would just keep the same vehicles in charge, to avoid the cost of routing vehicles in and out of chargers. The last inputs required, are the number of pickup and drop-offs per zone over each step over the horizon, and the average cost for going from one zone to the other.

Algorithm C

After obtaining results of algorithm B, the operator knows that it is best, for example, to re-position k vehicles from zone i to zone j and send q vehicles zone i to charger zone m. Knowing the location and the expected arrival time of vehicles at their last drop-offs, the operator can choose the best vehicles to fulfill the recommended flows by algorithm B. This is a job for the MILP optimization, algorithm C. Algorithm C will try to minimize the routing cost of going to charge and relocation trips. It makes sure charger utilization will follow as planned, choose the least charged vehicles for going to charge.

Algorithm C takes potential vehicles, that can satisfy flows obtained from algorithm B as input. For each pair, SoC of the vehicle at the last drop-off, EAT at potential destination point, and estimated consumption to the potential destination point is passed on as input. Further inputs include EAT of vehicles on their way to charging stations[1]. Naturally, the flows, and the total number of in charge vehicles are passed down from algorithm B.

[1] Charging trips that were previously decided on but they have not yet arrived

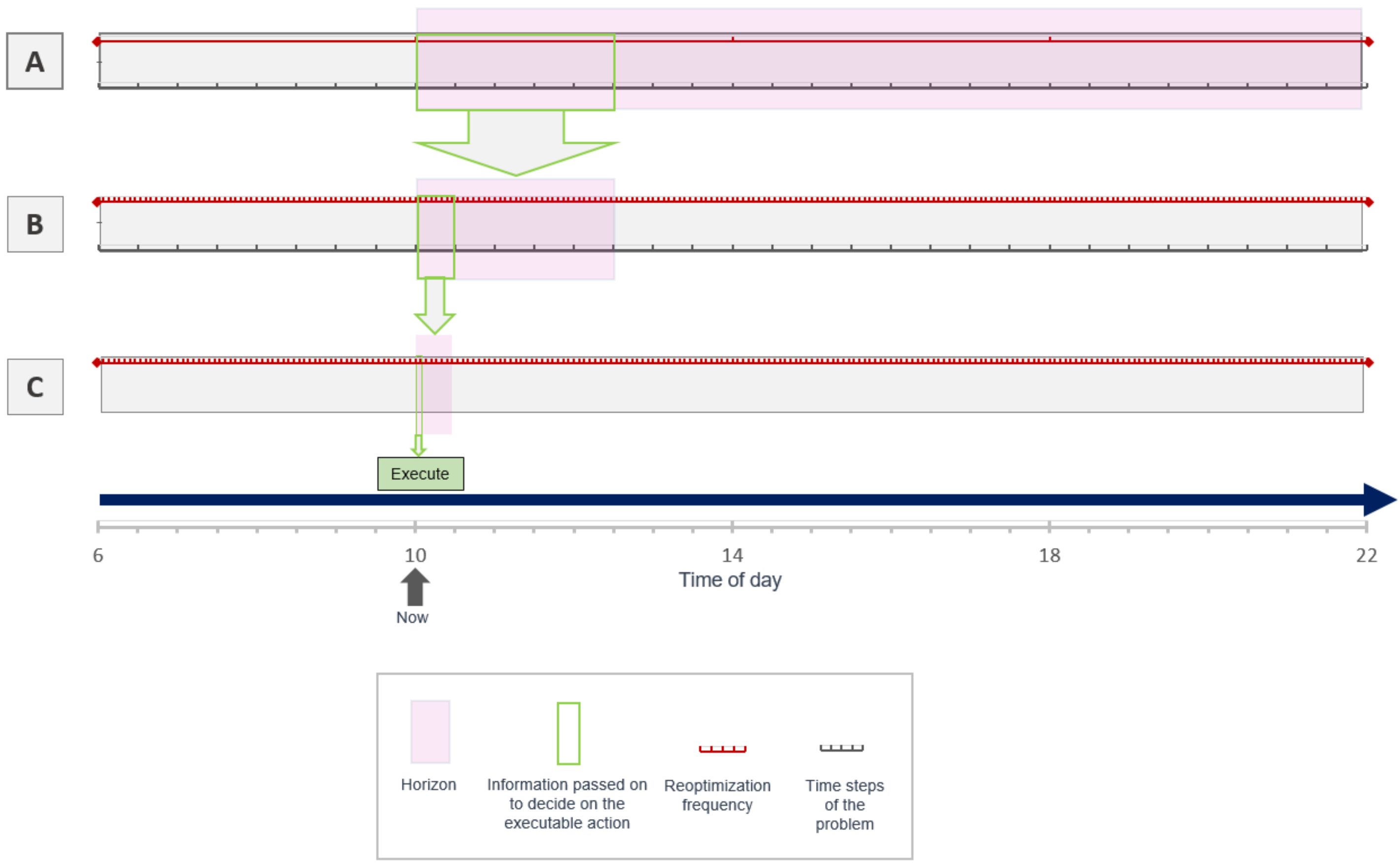


Figure 2: Time-line of the algorithms A, B, and C

The first level, algorithm A, only decides on time to charge. Its time steps are 30 minutes. Its horizon is until the end of the operational day. It will be re-optimized every four hours throughout the day. The second level, algorithm B, has a horizon of about two to three hours, it is re-optimized every four minutes. It will decide on the aggregated level where vehicles of each zone would charge or relocate to. The third level's (algorithm C) horizon is as long as the latest drop-off of all vehicles, and is re-optimized every four minutes. In all the levels only the decisions that have to start before the next reoptimization time will be executed (Figure 2).

0.3.2. Assignment extension

Algorithm R is used to select a vehicle among a restricted set received from the dispatcher. The first and foremost job of algorithm R is to avoid accepting the requests that will make it impossible for the vehicle to safely get to charging station after servicing its passengers. Therefore each time we have to compare the current SoC to combined consumption of the new tour that vehicle has to execute, and the trip from last point of the tour to its nearest charging station. As it is computationally expensive to find the closest charger every time a request is made, we pre-calculate the areas in the city, from which vehicles can get to a charger within 5, 10, and 15 kilometers. Then when the request is made we only need to check if the point of interest (last drop-off of the options) falls within the area of 5, 10, or 15 km (maximum) distance to the charging station.

After eliminating the unfeasible options, the rest of the k options are ranked based on three criteria. First, insertion cost of adding the request to the existing tour of each vehicle, which is based only on the distance. Second, for vehicles that reach an SoC of below 15% the cost of going to charger after the last drop-off is also considered, to avoid sending vehicles to areas far away from the charger, when most likely they have to charge. The last criteria is the higher the SoC of a vehicle the more likely it is to get the request. The core idea for the last criteria is that using the high SoC more frequently will result in a homogenized charge among the vehicles (e.g. having two vehicles with SoC of 30% rather than one with 60% and one 0%). This can allow more requests to be satisfied.

0.4. Performance

Experiments to report performance were run on a computer with 8 Intel R processors running at 3600 MHz using 32 MB of RAM, running Windows version 10. The performance was tested for fleet sizes of 150, 1500, and 15000. The gap for algorithm A is unknown due to using Column Generation heuristic [16]. The gap tolerance of algorithm B is 10%, and algorithm C is always optimal. Table 1 suggest that algorithm A, and B solving time remains independent of fleet size, and solving time grows linearly for algorithm C. The times to solve algorithm B and C are interdependent, and rely on the size (and consequently number) of the zones.

Table 1: Performance of algorithms with respect to fleet size

Fleet size	Algorithm A solve time (seconds)	Algorithm B solve time (seconds)	Algorithm C solve time (seconds)
150	499.86	1.596	0.005
1500	453.17	0.102	0.046
15000	462.47	0.163	-

0.5. Results

The results of the developed charging is compared to the base scenario, which has a simplistic approach to charging, also referred to as lazy charging for the remainder of the report. In lazy charging algorithm vehicles only go to a charger if they reach SoC below 20% and will then recharge in the closest charging station up to SoC of 90%. The vehicles will not go to the second closest charger if the first choice is full.

Figure 3 shows how SoC of fleet evolves in the day. The 150 vehicles start with full battery, and before the first charging trips, SoC of vehicles decrease homogeneously. SoC of vehicles going to charge is shown in orange and SoC of vehicles coming out of charge is shown in green. The vertical distance between the green and orange approximately shows the charging duration throughout the day. The algorithm is good at choosing low SoC vehicles to go to charge, which does in return increase the average cost of charging trips. Figure 5 shows that in the early hours fleet keeps up well with the demand, and as demand increases the number of satisfied trips reaches a saturation level. Starting from time-step 26, requests get rejected because of not having enough charge. We can also observe a slight decline in trips satisfied around time step 15, which coincides with the peak charging time of vehicles. The pink line in figures 7 and 9 show the daily plan by algorithm A. The figures compare the plan of algorithm A to the outcome of the the simulation. There is a delay in reaching number of in charge vehicles planned by algorithm A, as both algorithms A and B assume that vehicles can immediately arrive at the charger.

Figure 6 shows that in lazy charging operator starts rejecting customers for not having charge from time step 20, almost 2 hours earlier than in the smart charging, and a significant gap emerges between blue and orange line towards the end of the day. In the last time steps, in Figure 4, we see that many vehicles have battery above 50%, either because they charged to 90%, or they were not used from the start, while 75 vehicles accumulate in the bottom of the figure, with SoC less than 15%. As Figure 8 shows, lazy charging start sending vehicles to charge the same time it first begins to reject customers for not having charge. It makes no priority between the fast or slow chargers. It fails to use the chargers that were further away from the majority of low charged vehicles.

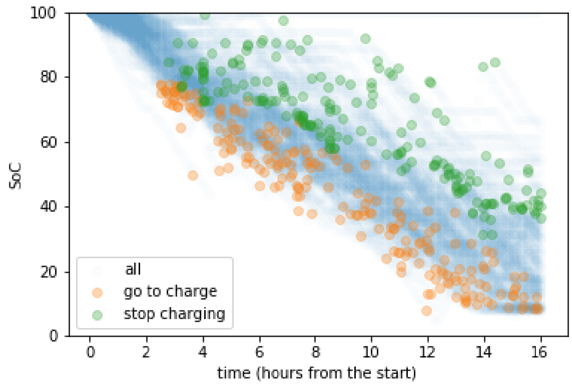


Figure 3: SoC of vehicles in time

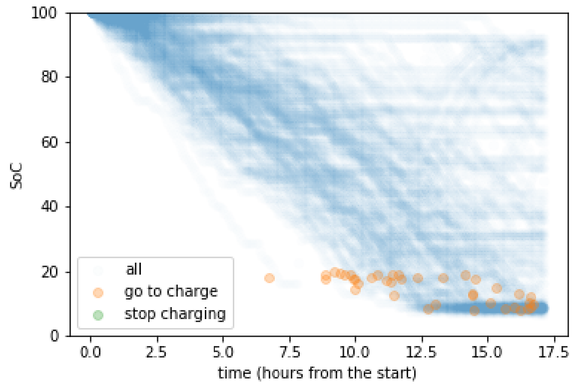


Figure 4: SoC of vehicles in time (lazy charging)

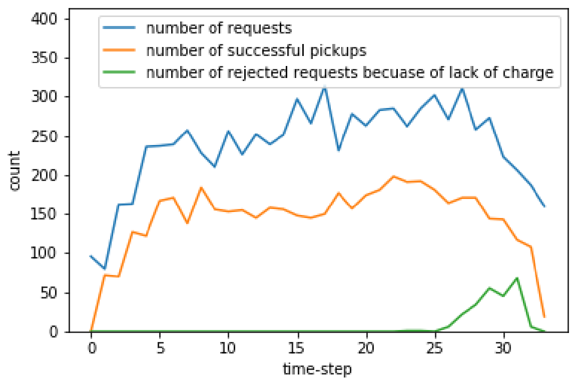


Figure 5: Trips

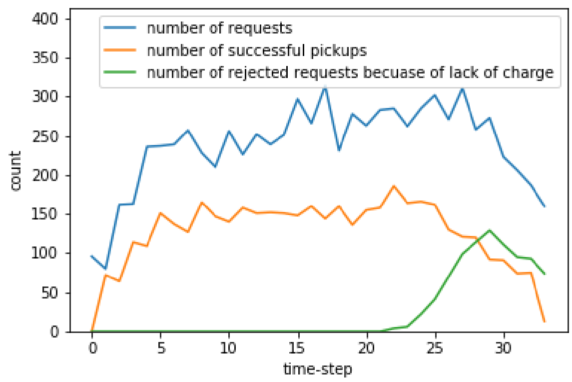


Figure 6: Trips (lazy charging)

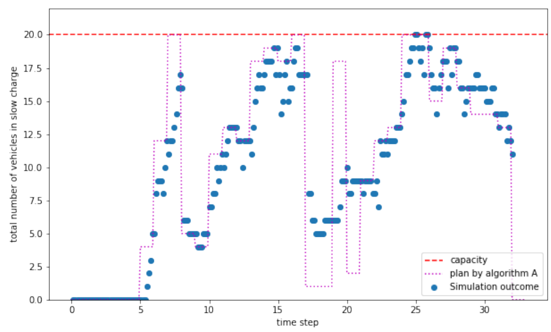


Figure 7: Number of vehicles in slow charge

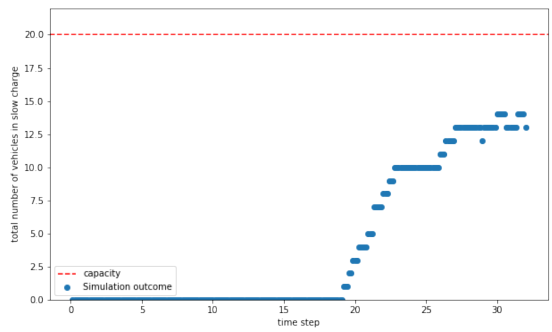


Figure 8: Number of vehicles in slow charge (lazy charging)

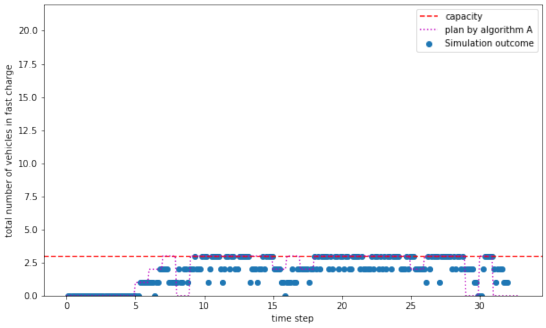


Figure 9: Number of vehicles in fast charge

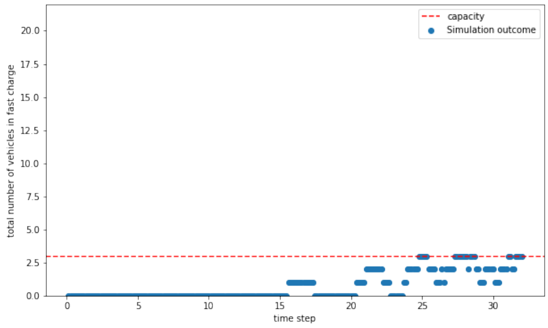


Figure 10: Number of vehicles in fast charge (lazy charging)

Table 2 shows that the presented charging algorithm, can satisfy around 600 more trips (8% higher acceptance rate) than the lazy charging algorithm, while spending far more on the trips to the chargers. In smart charging scenario, 7% of all distance traveled is for going to chargers and back and 5% of all distance is for relocation trips. Lazy charging only invests 0.5% of total distance on trips going to charges. This means with using smart charging, operator added 14% to its revenue while increasing the cost of an average trip by 16%. The designed charging algorithm can very well utilize charging stations at close to 100% rate by replacing vehicles in time, without any queuing, unlike the lazy charging who could only use 75% of the slow chargers at the time where more than half of fleet were out of charge.

Table 2: Results

	Unit	Lazy charging	Smart charging
Total trips	count	7740	7740
Rejected for not having charge	count	702	238
Percent accepted	percent	55.40	63.37
Distance per request	km	3.77	4.17
Ratio total travel time to sum of direct travel time	ratio	1.25	1.46
Number trip to charger	count	32	213
Avg trip distance to charger and back	km	2.1	7.0
Total charge gained	in ratio to one full battery	19.4	46.2

0.6. Conclusion

Key findings are as follows:

- The proposed algorithm addresses many shortcomings of the literature on operating electric taxies with dynamic requests. Including:
  - Charger capacity restriction
  - Limited demand [2]
  - Flexible charging duration
  - Consideration of the cost of going to chargers both in determining frequency of charging trips and which chargers to use
  - Drop in charging rate after 80% SoC
  - Considering SoC of vehicle while assigning vehicles to requests
  - Dependency between the relocation problem and the charging location problem
- Performance of the algorithm is within acceptable range, less than 10 minutes for daily planning, and less than 10 seconds for online planning. The performance of daily planning in independent of fleet size, and for online planing it grows less than linear.

[2]Limited demand implies that the profit associated with having the first vehicle in a zone is not the same as the tenth or the fiftieth vehicle in that zone.

- Planning charging of the fleet with horizon of a full operation day, proved to improve number of satisfied customers.
- The costs for an average trips were increased with use of smart charging, however the net gain for the operator remains positive.
- The smart charging algorithm can respond well to sudden changes, such as drop in temperature, change in electricity prices, and malfunction in the charging infrastructure.
- If inputs to the daily planning algorithm are inaccurate (prediction of total demand is wrong), it is not possible to fix them through the day, with the current set up of algorithms B and C.
- Tuning parameters of the three optimization algorithms can prove burdensome in practice.
- Expanding the model to include more complexities is hard.
- The algorithm developed can work jointly with variety of dispatching algorithms not just the one of PTV's. With some modification, it becomes applicable to range of services, such as on demand public transport. The important assumption that should hold is that the operator must own the fleet, or be in charge of the assignment and the charging decision. Therefore, it is not applicable to services such as Uber. It also can't be used as recommendation system for charging, as it cannot model the compliance of drivers, their adaption through time, or any competition between them.

Contents

List of Figures

List of Tables

1
Introduction

This chapter opens with the motivation behind the research (Section 1.1), and then it goes to describe the full scale problem of operating an electric fleet (Section 1.2), introducing all the sub-problems. It includes describing the system, the costs, and the constraints. The section continues to define the scope of the problem tackled in this study, that is the recharging aspect of the operation (Section 1.3). It will specify the main point of focus, and sub-problems that are left out or are out sourced. Next, the simplifications are discussed, including sources of stochasticity and variability that are ignored. Finally, we formulate the abstract problem that needs to be solved, and emphasize the qualities that the targeted solution should possess.

1.1. Research motivation and context

The persistent growth of urban population poses environmental, economic and social challenges to transportation systems that move the people, goods and services of society. Urban mobility has traditionally been restricted to privately owned vehicles or public transport. Advancements in technology and ICT have introduced innovative dynamic transit services that are radically changing the traditional views of the transportation industry, the social environment, and the business world. Emerging transportation network services (e.g., Uber, Lyft, Curb and Zipcar) have resulted in operational disruption and innovation, which enables the sharing of vehicular resources via a variety of mobile apps, ultimately shifting urban mobility modes from private vehicles to the dynamic transit [3]. Finally, the most recent trend, being the Mobility as a Service (MaaS), combines different transport modes to offer a tailored mobility package, similar to a monthly mobile phone contract and includes other complementary services, such as trip planning, reservation, and payments, through a single interface [10].

Electrification of the mobility services offers additional environmental benefits from decreased urban emissions, to higher energy efficiency. Operational cost are also lower, because of cheaper energy, lower energy expenditure per kilometer (as they can restore charge while driving downhill), lower repair costs, and longer lifetime, as number of moving parts in a car decreased by orders of magnitude [2]. Higher initial investment, for a lower operational cost, makes them especially interesting for mobility operators compared to private owners; Higher vehicle utilization enables the operator to compensate faster for the extra margin on initial cost. The same also holds for the aforementioned environmental benefits, in Tokyo for example, while only 2% of vehicles are taxi cabs, they contribute to 20% of the total carbon dioxide emission amounts [12].

Despite the environmental benefits of a transition towards zero-emission vehicles, the diffusion of electric vehicles (EVs) faces barriers such as range limitations, limited charging stations, long charging times and long queuing times. The driving range is even more restraining in hilly regions, in winter temperatures and when the driver wants to run heating and air conditioning. While private owners can rely on overnight charging, for their daily use, taxi drivers are not viable without at least charging once during the day.

A long term study on charging patterns among E-taxies (ET) in Shenzhen [20], which has the largest number of E-taxies in the world, reveals interesting findings on charging time and location preferences of drivers, and points out disadvantages such as capacity losses at charging stations, resulting from lack of full information, and sub optimal planning by individual drivers.

They found four charging demand peaks in each day, i.e., 3:00-5:00, 11:00-13:00, 16:00-17:00 and 22:00-23:00, and that the temporal pattern did not change significantly over a course of five years. The first and third peaks are before the rush hours for picking up taxi passengers. They investigated the electricity prices during the four peaks, and found that all of the four peaks are in relatively low electricity price durations.

They further reported that 20% of charging stations accommodate 80% of charging demand in 2013 due to their convenient locations, e.g., most of them are located in downtown areas, where the possibility for ETs to pickup passengers after charging is high. They concluded that charging stations play a key role in large-scale ET promotion and even though enough charging points are deployed, the unbalanced temporal or spatial charging demand and supply reduce the efficiency of the overall charging network.

In consideration of the example layed out, for effective operation, there needs to be a software backend which tracks vehicles, their battery state of charge and customer ride requests to manage and balance the electric fleets. The charging planner should navigate the driver to the best charging location based on availability of chargers, remaining battery range, and overall demand for charging.

PTV Mobility as a Service (MaaS) Modeller is a software component portfolio that supports city administrations and MaaS operators from planning through an operation to the management of their MaaS processes. MaaS Modeller studies evaluate the appropriate parameters to allow MaaS and eventually fleets of AVs to become part of the overall transport eco-system. The service includes integrating, regulating and controlling MaaS in existing mobility-mix or third-party operating systems. It handles assignments from modelling and evaluating MaaS operations, to simulating, optimizing actual operations, controlling, and where necessary, integrating with a city or state's overall mobility platform.

MaaS Dispatcher is the module of interest in this study. The algorithm targets the "pickup and delivery problem" where a fleet of vehicles operates on a road network in such a way that it can serve requests with temporal constraints. The specific use case for PTV is the operation (or simulation) of an on-demand ride-pooling service in a city or larger area. The algorithm is provided through an API as part of the real-time dispatching solution "PTV MaaS Operator". It also constitutes the core of the strategic planning tool "PTV MaaS Modeller". In this study, the basic functionality of the *MaaS Dispatcher* is to be extended to include Electric Vehicle (EV) operations.

1.2. Operation of electric fleets

The system is defined by an operator who owns a fleet of electric taxies and earns revenue from satisfying requests that come in real time from passengers going from an origin to a destination. The operator has to assign a vehicle to each request within a short time span. It will be assumed that all of the passengers are willing to share their ride within reasonable detour, and waiting time. Vehicles have to be charged several times during the day since their driving range is limited by the battery. They can recharge in private stations run by the taxi operator, at the depot, or in public charging stations. Moreover, the operator might re-position vehicles proactively to spots where historically, requests are expected to appear. Setting aside the fixed (investment) costs, the operator faces two direct transactions in daily operation. One is revenue received from the passengers, second is the cost of recharging either at public station or electricity cost in private stations.

The operator is constrained by its resources and the demand. The size of the fleet and the number of available chargers are hard constraints in this dynamic. Vehicles can only operate if they have enough charge. Any shortage in the number of charged cars or wrong distribution of charged cars within the operating zone will result in the rejection of potential customers. Apart from lower revenue, this may decrease operator's reputation. Naturally, we cannot charge cars if we need them for operation, and we cannot use more chargers than we have. Therefore, the number of vehicles we can charge at any moment is constrained both by the number of vehicles and the number of chargers.

Adding to the complication of the charging planning, is the charging and consumption curves. The former is to account for the non-linearity of charging in time, and the fact that it takes more time to charge the final 20% (from 80% to 100%). The latter can be affected by almost any driving condition, driving speed, traffic jams, heating, air conditioning, aggressive driving, gradient, wind, and rain, etc.

The operator takes many decisions throughout the day to maintain service quality and maximize revenue. We can further categorize the decisions faced by the operator. The different decisions can be formulated either in one mathematical problem or be solved sequentially. The list of sub-problems is as follows:

1. Routing problem: Knowing the order of pickups and drop-offs for each vehicle. The routing problem outputs the sequence of arcs to be passed by each vehicle.

2. Assignment problem: The decision to assign available vehicles to the requests. The order of pickups and drop-offs, or in other words, where to insert the new request within the current route of each candidate vehicle is also within the scope of this sub-problem.

3. Charging time problem: The time at which each vehicle should go to charge.

4. Charging location problem: The station where the vehicle should recharge.
5. Charging duration problem: The time to allocate to the charging of the vehicle.
6. Relocation problem: Through the day the number of drop-offs and pickups in each zone is not balanced. While vehicles do not have passengers on board, they can relocate to other zones in order to have a higher likelihood of a successful pickup. The relocation problem is then the decision to move a vehicle when it is empty.

The impact horizon of every sub-problem is different and is a prominent factor in the decision making; This is because often the decisions that bring the most profit in short time, will be worst off in the longer horizon. This typically motivates the operator to prolong the horizon over which the problems are solved. A typical case of such situation in the charging problem is charging vehicles only when no other trip can be made (lazy charging) versus charging when the battery is not yet critical (e.g. half full). The latter option will result in more frequent visits to the charging station in a short time, while having no immediate profit; however, in the long term, it can prevent having to charge at demand peak hours so that the full fleet is available at peak time.

1.3. Scope

Sub-problems

This project will accompany a series of services by PTV namely *xroute* for The routing problem and *MaaS Dispatcher* for the assignment problem. The *MaaS Dispatcher* will assign a vehicle to the requests that are realized in real time. The request are processed one by one, in a first come first serve manner; a vehicle assigned to a new request can already have passengers on board, this means that *MaaS Dispatcher* is also deciding how passengers rides are shared. The *xroute* is used for routing between already known waypoints (pickup, drop-off points, or any specified coordinate), it is used by the *MaaS Dispatcher* and will also be used in the charging planner to get information about the distance of the vehicle to the charges. In order to not interfere with algorithms that already exist in PTV, ideally the charging problem should be solved independent of the routing and the assignment problem as much as possible .

Since requests are assumed to only appear in real time and their waiting time is short (10 minutes), operator cannot plan any tours for the vehicles exceeding their charging time, thus routing problem can be formulated independent of charging time and location. The routing problem is therefore completely left out, and outsourced to PTV *xroute* service.

The assignment module can only decide based on waiting times and detour costs of passengers, and it holds no information on State of Charge (SoC) of the vehicles. Assignment in terms of searching for vehicles to assign to requests is also out of scope of study. However, after the search for potential vehicles based on other metrics (e.g. distance, waiting time) is over, the choice of vehicle can be altered (for example taking the second best vehicle), to prevent vehicles from running out of charge. The problem declared in this study will therefore face decisions on charging time, location, and duration along with relocating vehicles.

The last declaration on the sub-problems tackled by the study, concerns the ride-pooling aspect of the operation. Unlike private services, vehicles do not become free and available for charging after each drop-off. Each vehicle can have a chain of customers before it is free again. It is assumed that the charging planner does not need to cut the chain of the pooled passenger to rush charging. This means that there are enough natural breaks (vehicle having no trip assigned to it) through the operation, to provide chances for charging. This assumption should be revisited for higher vehicle capacities, and longer pre-booking times for the requests.

Connection to PTV modules

The existing *MaaS Dispatcher* can assign vehicles to requests, under the condition that vehicles have an infinite battery. Therefore, two additional mediums are required to ensure smooth operation of the fleet. One is sending vehicles to charge, with a specified duration. Another is to prevent the *MaaS Dispatcher* from choosing vehicles for trips that they do not have enough charge for.Moreover, the *MaaS Dispatcher* currently does not re-position vehicles. However, because of the dependencies between the re-positioning problem and the charging location problem we need a way to also communicate the re-positioning trips.

The first means of communication is addressed with *MaaS Dispatcher* defining a special kind of trip request with no origin, just a destination, and a service time[1]. Once requested, the trip will be enforced after

[1]Service time is time the vehicle has to spend in the destination location. A service of one hour would mean that the vehicle would be blocked for one hour after arriving at the destination.

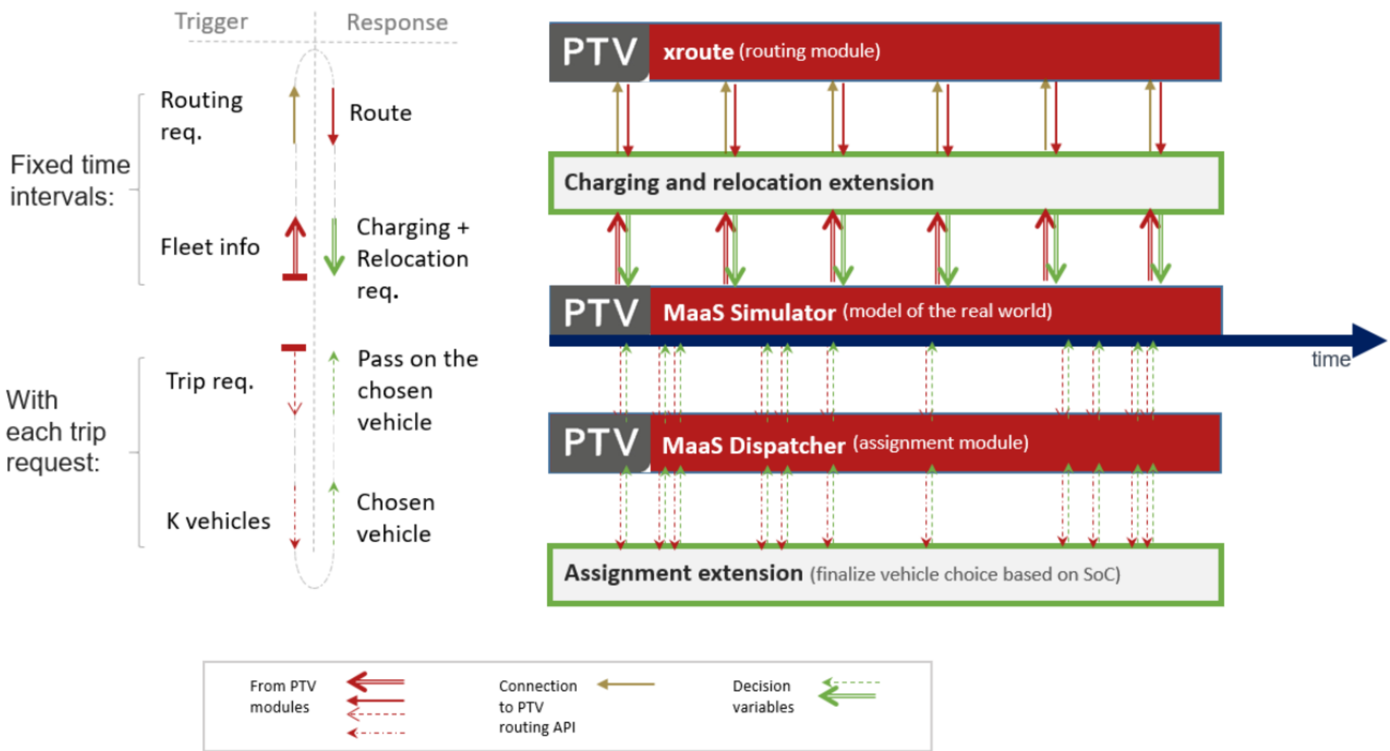


Figure 1.1: Connection with PTV modules

the last planned drop-off of the vehicle. Service time is greater than zero for charging request, and zero for re-positioning trips. This way the *MaaS Dispatcher* will always know the end of charging duration.

The chosen way to send the charging, and the re-positioning requests to the *MaaS Dispatcher* is for the simulator to call the *Charging and relocation extension* in fixed intervals. This way the charging and the re-positionings trips are decided on batches. Between these intervals, the *Charging and relocation extension* is blind to the new requests that may come in. By being called its information is updated, thus the *Charging and relocation extension* once more knows when each vehicle will be free again, and what their estimated SoC will be at their last drop-off point.

The second interaction point of the charging planner with the *MaaS Dispatcher* has to happen after each trip request has been activated. It is to guarantee that a vehicle has enough charge to make it to the closest charging location after its last drop-off. After each request is submitted, the *MaaS Dispatcher* will find k vehicles that can perform the trip, and return the k options to the *Assignment extension*. The *Assignment extension* can then choose the best one according to the vehicles' SoC level.

Figure 1.1 illustrates the interactions between the existing PTV modules, and the algorithmic extensions that are the products of this thesis. The *MaaS Simulator* is used to evaluate the case study, designed to test the effectiveness of the algorithms developed. It mimics the real world, namely the road network and the passenger requests. Four modules are employed to operate the vehicles cost effectively; they are provoked by the simulator as needed. All components from PTV are red in color; likewise, red arrows mean that they are constructed by the PTV components. The remaining blocks with green outline are products to be designed in this study. With each request the *MaaS Dispatcher* is actuated, once *MaaS Dispatcher* finds k best option for the request, the *Assignment extension* can finalize the option based on SoC of the k options. At fixed time intervals, simulator triggers the *Charging and relocation extension*, which has access to *xroute* through an API and can acquire as many routing requests as needs to decide if any vehicle should relocate to another zone or go to a charging station.

Charging facility

The operator only uses private charging stations both fast and slow chargers; hence there is no competition with exogenous vehicles for the charging spots. The distribution of the charging stations is non-uniform within the operating zones. The electricity prices are variable throughout the day, but they are known beforehand.

Fleet characteristic

More restrictive scenario parameters include homogeneity of the fleet, in terms of fuel type, capacity, consumption rate, driving range, charging compatibility. The charging curve is a piecewise linear function with a lower charging rate after SoC of 80%. The capacity for all vehicles is six passengers. The size of the fleet is also fixed throughout the day, currently, no breaks for the drivers are allowed. This also means that they would not disconnect and connect again to the cloud with a different SoC unpredictable by the operator. Finally, the consumption rate is assumed solely a function of distance for the case study, which is a stretch from reality. The implications and reasons for this simplification are discussed in Section 3.5. However, the methodology should remain generalizable to other consumption functions as long as they remain deterministic, and independent of traffic flow.

Information available

Planning for charging also requires the operator to have predictions for demand and means to estimate the consumption rate of the vehicles it owns. However, obtaining these is out of the scope of this thesis. Such data is used as input, with consideration for their limitations in quality.

Given that the passengers can share a ride, we cannot directly derive the number of vehicles required to satisfy a set of requests. Even if the set of requests is assumed to be fixed, given different scenarios for traffic state, the pricing scheme, the re-positioning policy, the policy to reject the unprofitable requests, and whether dispatching algorithm is passive or proactive chances of trips being pooled can significantly vary. On the other hand, the overall consumption of fleet, and thus the plan for charging, will depend on the operational data of fleet, and in particular the distance the vehicles travel during the day and not the number of requests. Therefore, for the remainder of the project, it is assumed that the operator either has a model to map the request data to operation data, or that we directly have historical data on the operation of the fleet.

The charging planner algorithm is expected to rely on an aggregated level of demand for planning charging of the fleet. The algorithm has access to a set of information including the number of active vehicles required through the day, aggregated per time-step (typically 30 minutes). Moreover, it is assumed the algorithm has reliable information on the number of pickups and drop-offs per zone per time-step. The actual requests, with time-stamp, origin, and destination only appear in real time, and the operator is assumed to have no prior information on them. The demand, or more precisely the associated operation time required can vary within days, and the charging planner should be responsive to that. However, charging cannot be optimized over stochastic or multiple demand scenarios per day.

Travel time variation

Travel times are assumed to be fixed throughout the day. Travel time variability affects the charging plan through three main channels. Firstly, longer travel time affects vehicle operations, vehicles are longer in use for the same amount of requests. As explained previously, this layer is out of the scope of this research. Second, travel times or the associated traffic state they represent affect acceleration patterns of the vehicles, and hence consumption per unit time would also be different. The final channel is that the same way the normal operation is more expensive in peak hours, going to a charging station will also cost more during peak hours. While the requests cannot be shifted in time, going to charge time could be shifted, if the operator had the relevant information. However as mentioned this is out of scope of this study.

Problem outline

Having shaped the scope, and having trimmed the real world complexities that lie outside of it, we can outline the abstract problem of dynamic recharging and relocating of electric fleets. The problem decisions, objective and constraints are as follows:

<u>Decisions</u>:

1. The decision to mandate a specific vehicle to go to a charging station after finishing its current tour. These decision can be made in fixed time intervals (e.g. 4 minutes)
2. The decision to mandate a specific vehicle to relocate after finishing its current tour. These decision can be made in fixed time intervals (e.g. 4 minutes)
3. The vehicle to be chosen for trip requests among a restricted set, given by the *MaaS Dispatcher*

<u>Objectives</u>:

1. Level of empty routing (going to charging station, re-positioning) is kept to a minimum
2. Low electricity cost
3. Serve as many passengers as possible

<u>Constraints</u> (including but not limited to):

1. No vehicle runs out of charge in the middle of the operation

2. Passengers waiting time is restricted
3. Fleet size is restricted
4. There are limited charging stations
5. Capacity of charging stations are restricted
6. Vehicle capacity is limited
7. Charging rate is restricted

Chapter 2 depicts a picture of the studies concerned with taxi operations and electric recharging. In Chapter 3, the algorithms for *Charging and relocation extension* and *Assignment extension* are developed. A case study in Barcelona is designed in Chapter 4, distinguishing the general set-up of the experiment, first to compare the algorithm with a basic charging rule, and the rest to test the algorithms under different conditions. The chapter presents the main results for each scenario, and draws a comparison between them. This is followed by a discussion on overall performance of the algorithms, and finally we conclude on the findings.

2
Literature review

Beginning with outlining the studies touching on operation of electric fleets, the chapter follows with identifying decisive differences in characterization of the charging problem in the literature, and common methods to tackle each category of problems. Most notable studies in each category are then described in further detail, along with discussing relevance of their methodology in regard with the problem laid out in this research. The chapter ends with a summary of the literature, and a conclusion on the basis it provides for solving the formulated problem.

2.1. Overview

The literature on the operation of electric vehicles can be divided into two main branches. The first and more consolidated version of the problem is an extension of the Vehicle Routing Problem (VRP). In this branch almost always the requests are known beforehand. The requests are usually satisfied in the adequate timewindow. This gives the operator flexibility to chain these requests with full information, where at the same time, this flexibility makes the problem computationally harder. Another crucial characteristic of such problem is that the richness of information about the requests allows routing, assignment, and charging problem to be solved at the same time. This is where the major difference between this study and most of the literature facing the static problem comes from.

In this study, the routing is outsourced, meaning that while the algorithm will have access to travel times or distance between any two points of interest, it does not engage in choosing the arcs that construct the route. This is aligned with the fact that there is no prior information on the exact location and time that requests appear. While VRP methodologies are less practical in solving the dynamic charging problem, studying the construct of problems in VRP, would enhance the understanding of the problem. An overview of dynamic VRP studies is presented in Section 2.2, followed by a more detailed description of the selected papers solving static charging problem.

The second branch of studies directly addressing the recharging of the electric fleet, deals with the more dynamic variant of the problem. They are less concerned with routing, and focus on exploiting the more aggregated information on the demand side. However, the number of distinctive problems in terms of assumption and focus, are as numerous as the studies done. Hardly any two studies solve the same problem.

One of the assumptions that can notably influence the methodology is whether the problem is constrained by the demand or not. Some studies assume that the profit associated with having the first vehicle in a zone is the same as the tenth or the fiftieth vehicle in that zone. The consequence is that in this case, the fleet does not have to be controlled centrally, any vehicle can choose the best action for itself, and that would be globally optimal. In this study, the problem is assumed to be constrained by demand. Consequently, the algorithm developed for this problem is unnecessarily complex if the fleet is much smaller compared to the available demand.

The other impacting assumption is whether the number of chargers is limited. Surprisingly, few studies address this issue. The implication of unlimited chargers is less binding constraints between the vehicles, hence the impact of a vehicle charging will not propagate in time, as much the as the case with charger limitation. This will also change the horizon required to solve the problem optimally. A summary of a selected static and dynamic studies is presented in Table 2.2. A more in-depth discussion of methods in dynamic operation of electric vehicles is presented in Section 2.3

2.2. Vehicle Routing Problem

2.2.1. Overview of problem and methods

The literature has different interpretations of the dynamic problem, in other words, there are different levels of dynamism. The level of dynamism refers to both the frequency of changes and the urgency of requests [9]. The problem at hand has the highest degree of dynamism in both measures, in a sense that there is no pre-booking the trips a day or an hour before. However, the literature starting point on the dynamic problem is dealing with minor changes to the requests that were entirely made a day before the operation. Most of the literature consist of problems where at least a portion of requests are known beforehand.

In most cases, the problem was approached, first by solving a static problem, after which insertion heuristics were used to adapt to the changes. Another share of so-called dynamic problems refer to the problems where request are fixed, but the travel times change in real time. In both cases, dynamic changes are made on top of a pre-planned route. However when requests appear only online and have to be served as soon as possible, the Pre-planned routes are not available.

The vehicle routing literature with only online demands can also be divided into two broad categories, with and without stochastic information. In a dynamic and deterministic (DD) problem, no stochastic information about the future is known. For instance, nothing may be known about the location of a customer until that customer requests service; Or, nothing may be known about the quantity to be demanded until when that information is revealed. As a consequence, exact methods only provide an optimal solution for the current state, but do not guarantee that the solution will remain optimal once new data becomes available. Therefore, most dynamic approaches rely on heuristics that quickly compute a solution to the current state of the problem [15].

Dynamic and stochastic (DS) routing problems can be seen as an extension of their deterministic counterparts, where additional (stochastic) knowledge is available in the dynamically revealed input. Approaches for this class of problems can be divided into two categories: those based on sampling and those based on stochastic modeling. As their name suggests, sampling strategies incorporate stochastic knowledge by generating scenarios based on realizations drawn from the random variable distributions. Each scenario is then optimized by solving the static and deterministic problems. On the other hand, approaches based on stochastic modeling integrate stochastic knowledge analytically.

Common methods in Stochastic modeling are Markov Decision Process (MDP), Approximate Dynamic Programming (ADP), adaptations of Linear programming, waiting strategies (which mostly apply to the delivery problem with time windows). In dynamic models for transportation operations [14], Powell argues that the biggest challenge of using distributional forecasts is the lack of effective tools for solving problems under multiple future scenarios. By contrast, we do not have any difficulty using distributional forecasts when we use value functions.

In dynamic programming, decisions reflect what we know and the impact of our decisions on other parts of the problem. It is significant that a dynamic-programming based approach, which uses value functions to capture the impact of decisions made now on the future, incorporates uncertainty relatively easily. The effect of different possible outcomes is captured in the value function, which is much simpler than solving a problem at time t with an explicit set of multiple scenarios. Specifically, a decision function that uses value functions is implicitly using a forecast of exogenous outcomes, expressed through the value functions. Adding value functions to a decision function is equivalent to using a forecast of the impact of a decision on another agent.

Proximal point algorithms are another method where future estimations can be enforced to objective function through past optimal decisions, or decisions which would have been optimal for past operations. The idea is that on an aggregated level, it is intuitively reasonable to make decisions that do not deviate from a plan by too much. This is formulated by adding deviation from aggregated previously optimal decisions to the online cost function.

Table 2.1: Common methods in variations of Vehicle Routing

VRP literature		Static	Semi-Dynamic	Dynamic
		Not urgent requests. (No-reoptimization)	Not urgent requests. Part of requests known.	Urgent or unurgent requests. Requests are not known.
Deterministic (Passive)	Optimization only over known requests.	**C**olumn generation. **C**lustering methods. **L**ocal search.	**M**ostly depend on resolving static problem. **M**ultiple plan approach.	**M**ostly depend on resolving static problem. **C**urrent PTV dispatcher. **M**ultiple plan approach.
Stochastic (Pro-Active)	Something about the requests are known at the time of optimization.	**R**obust linear programming. **M**emetic algorithm. **I**nteger L-shaped method. **R**ollout policies.	**M**DP[1] **S**ampling **P**roximal point	**M**DP^h **S**ampling **V**alue function approximation **P**roximal point

2.2.2. Selected studies solving the Electric Vehicle Routing Problem

[18] adapted the Green VRP to EVs and added time window constraints, introducing the Electric VRPTW with Recharging Stations(E-VRPTW). The aim was to find tours satisfying charge constraints and time window constraints. The problem is solved by a Variable Neighbourhood Search (VNS) approach using Tabu Search (TS) as a local optimization technique. This is the example of full information, full day optimization of combined sub-problems, where customers can be chained effectively together, since they are not urgent. The main challenge in this category is keeping the problem tractable, managing information on the other hand remains uncomplicated.

Another example of full information problem is [17] where they solved the vehicle routing problem, defined by a set of customers that have to be served by a mixed fleet of vehicles composed of an heterogeneous fleet of EVs with distinct battery capacities and operating costs. Charging stations offer charging with different technologies, time-dependent charging costs are subjected to operating time windows. The objective was to minimize the number of employed vehicles and to minimize the total traveling and charging costs. The developed multi-start algorithm was based on the Iterated Local Search metaheuristic which uses a Large Neighborhood Search with two different insertion strategies in the Local Search procedure.

[3] focused on mid-tour charging behavior in a small time scale and serving interactions among a limited set of EVs with coupled constraints. In the horizon of the problem, requests were pre-booked with no time windows, typically for about one to four customers per vehicle. Their objective had three terms, route distance, passenger travel time, and charging cost. The dynamics of the system were modeled with Mixed Integer Quadratically constrained programming, which was solved using a commercial solver. While this is closer to the problem faced in the current study, requests are not as urgent as in this study. The lack of potential to scale to large fleet size is another major point of contrast.

2.3. Studies solving dynamic charging planning

2.3.1. Studies without constrained demand

[12] proposed a decision making flow for the fleet of E-taxies based on two criteria, the future taxi demand at the destination (labeled as low or high), and availability of the charging station at the destination (labeled as low or high). The possible outcomes are dispatching the lowest or highest charged vehicle to the request. To ensure that vehicles do not run out of charge, they conduct a reachability analysis based on historical data of the average consumption of each driver. This problem in fact, has the same level of dynamism as the current

[1] Markov Decision process

study and tries to incorporate the information on SoC of the vehicle into the assignment problem in a simple way. The disadvantage is that they do not incorporate the routing cost while selecting a vehicle for a request.

[19] have solved the problem for one vehicle with Markov Decision Process. The state of the vehicle was defined by its location (nearest junction), SoC (discrete levels), and time (with one minute increment). Actions possible to drivers were waiting, relocating and charging. Using New York taxi data for conventional vehicles, they pre-calculated probabilities for successful passenger pickups for all junctions and time slots. Also the probability of passenger commuting from the origin junction to the destination junction. The transition function was adapted for electric taxies; to do so, they assigned a different probability for each SoC category, by translating the SoC to the reachable area from the origin junction. The algorithm looks only at the time to charge problem and always chooses the nearest charging station. The problem is solved by starting from the last time slot and working backward to the beginning time slot. For example, to solve the optimal policy for a morning shift, the algorithm starts to solve the maximal expected net revenue at the end of shift, and works backward. The first barrier to using this method is, as mentioned, that actions of one vehicle do not affect the others. The second barrier is that the method indeed needs an exceptionally rich data set, to estimate such detailed transition function, without generalization.

[22] also solves the problem for individual taxi decisions, with a focus on highly stochastic charging prices. The optimal charging sequence was determined by a thresholding method. That is, a sequence of thresholds are computed in advance; then, at each time slot the threshold is compared with the announced real electricity price to determine whether to charge in this time slot or not. Cost of driving to charging station and incomes through the day was assumed fixed. Their work was extended in [23] to sequential solving for time and location of charging. After an EV decides to charge at a slot, the driver still needs to choose a charging station from the surrounding ones, which associate with different traveling time and queuing time. It is notable that many EVs in the same region request to charge at the same time slot, and an EV's individual selection of CS may affect the queuing time of other EVs that arrive later at the same CS. Thus, the spatial selection problem turns into an interaction among multiple EVs. To tackle this technical challenge, a game theoretic approach was proposed to solve the spatial selection problem in a distributed way. That is, every driver will make his own selection and Nash Equilibrium (NE) can be achieved through iterations.

In a recent paper by [1] approximate dynamic programming was compared to a myopic policy in a joint car-rider allocation, car relocation, and recharging problem, for a ride-sharing system, using autonomous fleets of electric vehicles. In the Dynamic Programming approach they took into account the future trip demands; however, under a fixed demand scenario. They used a look-up table value function representation. The value function was defined over discrete states that capture time, location and battery charge level. They assumed that value function is linear with the number of cars. Value of a single car in value function literary means that if we drop a car in a zone a_1 and battery level a_2, at time t, then the marginal contribution of that car from time t up to the end of the simulation horizon T, would be approximately equal to this value. Then each decision is made by comparing sum of the immediate profit from that decision and the value of the resulting state of the system, after that decision is executed, which is approximated by the value function. Approximate dynamic programming is good with dealing with demand stochasticity, when no posterior information on demand is available. However, it cannot account for higher average demand, since value function cannot hold demand as a variable.

2.3.2. Studies with constrained demand

In an autonomous and electric car-sharing context, [8] proposed a model to optimize transport service and charging at two different time scales by running two Model Predictive Control optimization algorithms in parallel. Charging is optimized over longer time scales to minimize both approximate waiting times and electricity costs (30 min time step, and 5 hr horizon). Routing and relocation are optimized at shorter time scales to minimize waiting times (2 min time step, and 30 min horizon). The results of the long-time-scale optimization act as charging constraints in the short-time-scale optimization. This approach allows efficient optimization of both aspects of system operation. The problem is solved as a mixed-integer linear program. There are two main differences between application in [8] and the current study. First, because in car sharing the stations themselves hold the charging plugs, there are no separate trips just for charging. Second, they assumed that unsatisfied demand in one time slots propagates to the next time slot and will not be rejected, while this is not the case with strict maximum waiting time in this study.

In [25], in an offline procedure, they simulate the requests one day ahead without charging to get the energy demand of the fleet through the day. Then they optimize time to charge problem with variable charging prices. In the online optimization, EVs are grouped together. For each group, number of demand, and number of fully charged vehicles are calculated. Then each group has to decide what ratio of vehicles it will allocate to transportation and the rest will charge. In order to decide, a utility function is defined for each group, which decreases with charging costs and squared difference between the number of transportation EVs and demand for EVs in the group. Overall, the fleet is constrained to have more transportation vehicles in total than total demand. In addition, the number of the total in charge EVS is equal to the amount planned by offline planning of charging. The online problem is solved with a cake cutting game[2], in which they proved an equilibrium exists. The main limitation is that they do not associate a cost with trips where EVs go to charging stations. There is no limit on the number of chargers, as they are assumed immediately available. Furthermore, there is no explicit mechanism to prevent individual vehicles from going out of charge. It is simply trusted that under the offline charging planning they will not go out of charge.

2.4. Conclusion

In conclusion studies using Markov Decision Process, and Dynamic programming (e.g. [19], and [1]) are best at incorporating operational and demand stochasticity, through use of transition function, and value function, however, they fall short in modeling constrained demand, and different demand scenarios, where subsequent information is available compared to base scenario for which they were calibrated for. The main take from [25] and [8] is the need to plan charging over a long horizon. [8] uses larger time steps for planning time and location of charging, while keeping it at individual vehicle level. [25] only plans for number of vehicles that charge in each time step, and does not tie it with location of charging or individual vehicles. The latter is preferred, as planning for time and location of charge five hours ahead under demand stochasticity, raises computational complexity, without providing benefits. In regard with dependency of assignment problem on SoC of vehicles, [12] imposes a fixed rule to choose high SoC vehicle, if destination has no charger available, [12] always chooses highest SoC vehicles for operation, and lowest SoC vehicles for going to charge, and [8] is flexible, as SoC information is included in the assignment problem. Therefore, this study will investigate if including SoC of vehicles as a deciding factor in the assignment of vehicles to requests, would improve profitability.

[2] In the cake cutting game, each vehicle groups gets a piece of transportation and charging cake, based on how much they value the piece

Table 2.2: Methods in Electric Vehicle Charging Problem

		This study	[5]	[12]	[11]	[17]	[3]	[18]	[13]	[21]	[24]	[8]	[1]	[19]	[25]
problem	Limited demand	✓				✓		✓	✓	✓		✓			
	Limited chargers	✓	✓	✓			✓				✓				✓
	Dynamic requests	✓		✓	✓						✓	✓	✓[a]	✓	✓
	Prebooking					✓	✓	✓	✓	✓		✓			
	With-repositioning	✓								✓		✓	✓	✓	
	Pro-active	✓									✓	✓	✓	✓	✓
	Variable charge pricing	✓	✓			✓	✓			✓	✓	✓			✓
	Stochastic charge pricing									✓	✓				
	Variable travel time						✓							✓	
	Time to charge		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Where to charge	✓		✓		✓	✓	✓	✓	✓	✓	✓			
	Partial charge	✓				✓	✓	✓	✓			✓		✓	
	Consumption function of Distance	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
	Linear charging curve			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Application	Ridesharing	E-buses	E-Taxi	Ridesharing	VRP[b]	PDP[c]	VRP[b]	PDP[c]	E-bus	E-taxi	Carsharing	Ridesahring	NY taxi	E-Taxi
method	Optimization	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Method	MIP[d]	MIP[d]	Heuristics	Heuristics	MIP[d]	MIQCP[e]	MIP[d]	MPC[f]	MDP[d]	thresholding + GT	MPC[f]	DP[g]	MDP[h]	Cake cutting game
	Method sub class					local search	VNS[i] /TS[j]	Clustering/ VNS[i]/ TS[j]/ CG[k]		AVI[l]	VI[m]				
objective	Energy cost	✓	✓			✓	✓	✓		✓	✓	✓	✓		✓
	Revenue	✓				✓	✓			✓	✓	✓	✓	✓	✓
	Waiting time (passenger)	✓					✓					✓			

[a]with a fixed average
[b]Vehicle Routing Problem
[c] pickup and Delivery Problem
[d] Mixed Integer Programming
[e]Mixed Integer Quadratic Constrained Programming
[f] Model Predictive Control
[g]Dynamic programming
[h] Markov Decision Process
[i]Variable Neighbourhood Search
[j]Tabu Search
[k]Column Generation
[l]Approximate Value Iteration
[m]Value Iteration

3
Methodology

The first part of this chapter explains the motivation behind the decision on methodology. It then presents the break-down of the problem into four sub-problems, and shows how the components are designed to interact with one another. Finally each component is described individually in Sections 3.2 to 3.4.

3.1. Algorithmic outline

Defining factors in choice of methodology

The choice of methodology is promoted by the necessity to achieve balance between two properties of the problem, while respecting two practical constraints. The first property is the impact horizon of each sub-problem, and the second one is the information available over that horizon. The first practical constraint is computational complexity for solving over the associated level of detail. Finally, the last constraint is that the output format of charging planning (in terms of the type of decision and the time it should be enforced) should be compatible and communicable to the base *MaaS Dispatcher*.

Impact horizon vs available information: Time and duration of charge

The time and duration of charging a vehicle can impact its operation over the next six, to twelve hours. At the same time, if one vehicle is charging, it means that another vehicle cannot charge, which will impact the operation of other vehicles as well. This requires the charging time and duration optimization to have a long horizon. However, the operator does not know the location of the vehicles one hour ahead of the decision, or whether a vehicle would be free, and consequently the operator is not aware if the vehicle is available to go to charging. The information available over the required horizon is only reliable[1] in the aggregated form, that is the total number of vehicles active. Therefore, for time to charge and duration of charge we need to make aggregated decisions over a long horizon. The decision can be the number of vehicles that have to start charging at each time-step, until the end of the operation, along with their average charge duration.

A similar strategy was used in [25]. However, in the model for their offline time to charge planning, they assumed all vehicles can use the same shared mega-battery. While that would make the problem easier to solve, it implies the assumption that charge over the fleet is almost homogeneous. The cost of having a homogeneous charge is paying more frequent and shorter visits to the chargers. This can work under a dense network of chargers, where the cost of going to the charger is negligible. However, if chargers are limited or not densely distributed, such an approach would by far underestimate the charge required by the system. To make this tangible, the reader can compare a case where 10 vehicles all have 30% charge, and a case where 5 vehicles have 60% and the other five have 0%. A shared battery would always tell us we can serve 10 trips, at the same time. The real operation with limited chargers is closer to the latter case, and thus at any given time, we must have more reserved charge[2] in the system, than if we were to assume we have a shared battery. The component that deals with time and duration of charge over the course of the day is called *Daily charging planner* and it uses algorithm A; and is elaborated on in Section 3.2. The *Daily charging planner* guides the component *Online charging planner*, that decides in real time which vehicle has to charge.

Impact horizon vs available information: Re-positioning and charging location

Two mechanisms may affect the chosen horizon for location of charge problem. In short horizon optimization, the operator might fill up a charging station with vehicles coming from zones further away, while later, vehicles would have been available from closer surrounding areas. As mentioned before, we do have

[1] There is more systematic pattern than there is randomness

[2] An excess charge that we cannot immediately access

information on the number of drop-offs per zone; Hence the operator can rely on this information to determine if in the near future, it can send vehicles to a charger without having to make costly empty trips. Another way the operator can profit from the number of pickup and drop-offs over the next two hours is not only to consider the inbound trip to the charging station but also the outbound (where the vehicle should move to after charging) while choosing a location for the vehicle to charge. This already hints that it is beneficial to have a horizon somewhat longer than the average charging duration.

The decision for re-positioning vehicles from one zone to the other also requires a horizon longer than an average trip length. Imagine taking vehicles away from a zone at time t based on the analysis that they were not needed at the moment. However, the operator might end up having to relocate back to the same zone in the next time-step. Thus keeping the vehicle idle in the zone would have been the optimal solution in this case.

We have two constraints on the available information over a horizon longer than an average trip length. The first is that the information on pickups and drop-offs per zone is aggregated, therefore we still do not know the location of any individual vehicle. This constraint on information can be addressed by dividing the decision into two steps. Step 1, making aggregated decisions on how many vehicles to move from any zone to chargers and to other zones, over the longer horizon; this would result in flow of vehicles between the zones. Step 2, for the horizon for which the operator has information about the location of vehicles, it can select which vehicles satisfy the flows between the zones. Therefore, what is called the *Online charging planner*, actually consists of two optimization algorithms B, and C, which will be explained in Section 3.3.

The second constraint on the available information is that we do not have SoC information for vehicles in a particular zone, in the longer horizon. SoC information is required because not all vehicles can serve trips (if they have very low SoC), and not all vehicles can go to charge (if they have high SoC). The constraint will be elaborated on in Section 3.3.2,

Connection between the algorithms

The dynamics between all mentioned elements is illustrated in Figure 3.1. The simulator represents the real world. At the beginning of the day, it will contact the *Online charging planner*, who will request a temporal plan for charging, from *Daily charging planner* (Algorithm A). Then using algorithms B, and C, it can inform the simulator if any vehicles have to go to charge over the course of the next four minutes. It will also update relevant parameters in *Selecting vehicle for request*. The details of these parameters in discussed in Section 3.4. From then the simulator will start processing requests one by one. Each time a new request comes in, *MaaS Dispatcher* sees if the request can be accepted, and will send k vehicles who can service the request to *Selecting vehicle for request* (Algorithm R), then algorithm R will return the best vehicle based on their SoC. When four minutes have passed in the simulation, the simulator will call the *Online charging planner* again. The time-step specified in Figure 3.1 for algorithm A and B shows the discretization of time used in their formulation.

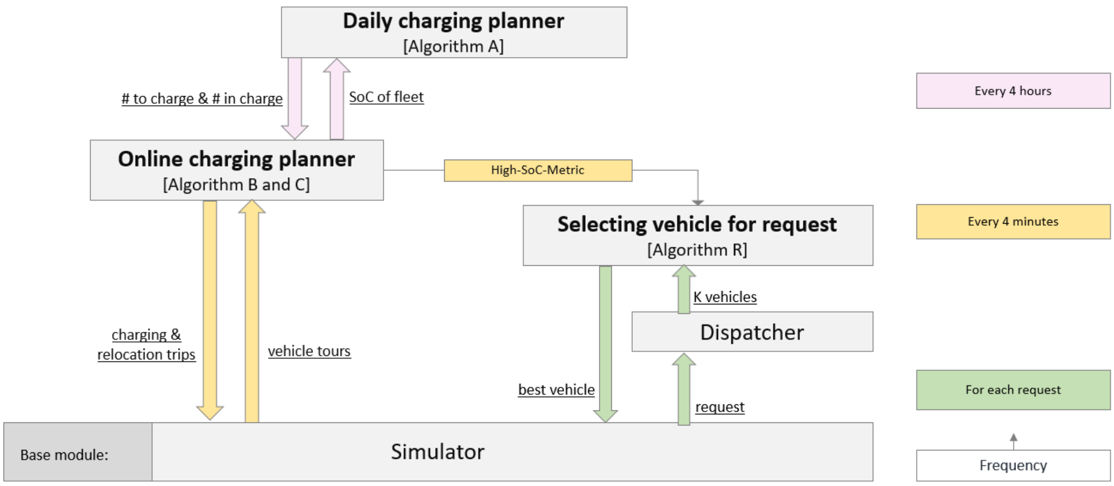


Figure 3.1: Algorithmic approach

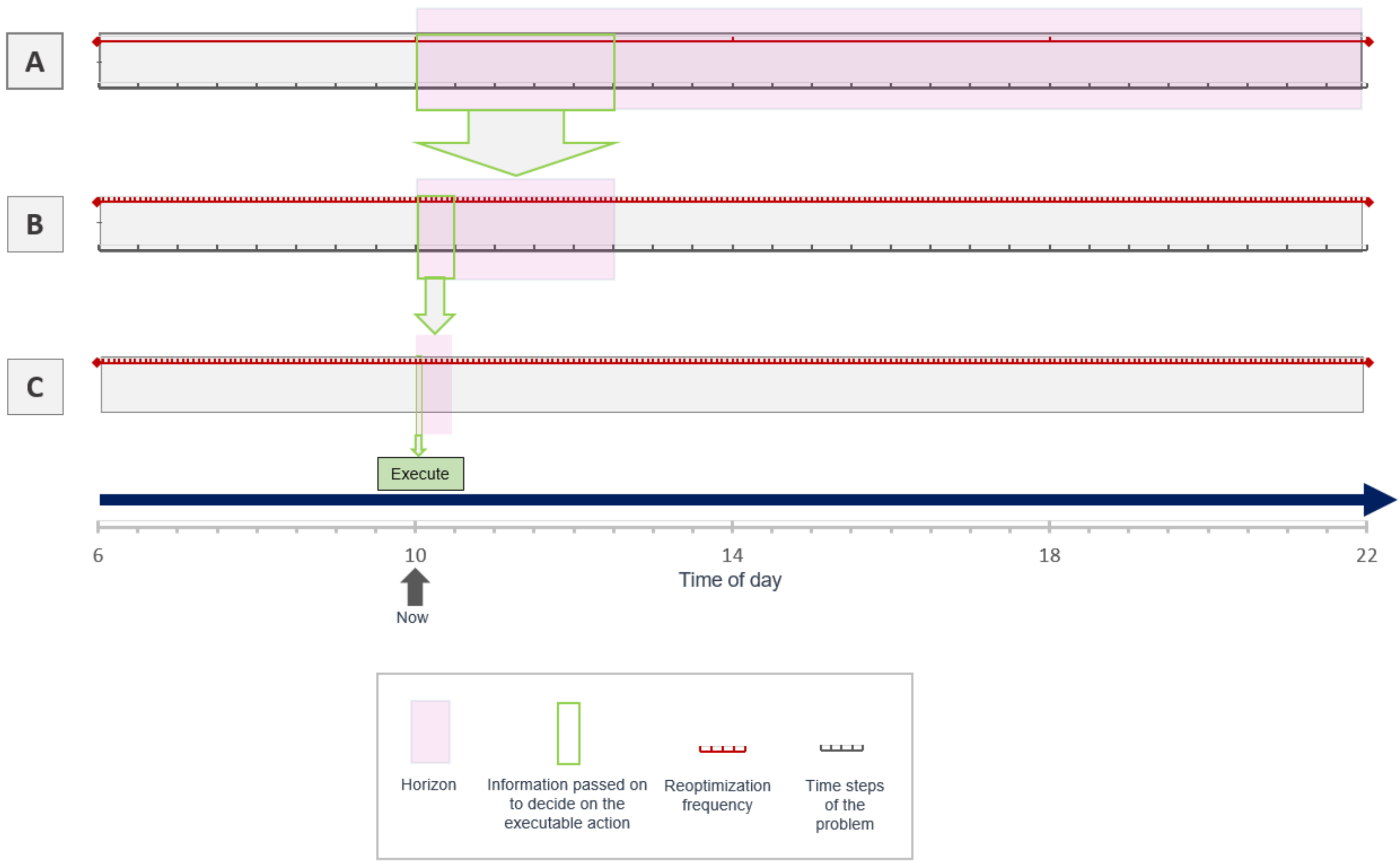


Figure 3.2: Time-line of the algorithms A, B, and C

Summary

Charging and re-positioning trips are the result of a three level hierarchical optimization. The first level, only decides on time to charge with steps of 30 minutes and horizon, until the end of the operational day. It will be re-optimized every four hours throughout the day. The second level has steps of 30 minutes and a horizon of about two to three hours, it is re-optimized every four minutes. It will decide on the aggregated level where vehicles of each zone would charge or relocate to. The third level's horizon is as long as the planned tours of the vehicles, and is re-optimized every four minutes. In all the levels only the decisions that have to start before the next re-optimization time will be executed (Figure 3.2). As will be shown in Section 3.2, algorithm A is solved by Column Generation, therefore, it is well scalable with respect to the number of vehicles. The size of the algorithm B depends only on the number of zones, therefore also scales well as fleet size increases. Algorithm C, however, will become increasingly hard with an increase in fleet size. The optimization itself should not impose a major delay, because of algorithm C's short horizon. However inquiring consumption cost (an input of algorithm C) for a growing number of options (vehicle-destination pairs) can prove challenging. An empirical analysis of algorithms' performance is presented in Section 4.2.

3.2. Daily charging planner (Algorithm A)

3.2.1. Modeling

A Mixed Integer Linear Program (MILP) is developed to represent the charge dynamic of the fleet. By charge dynamic, it is meant that dimensions representing one vehicle are reduced to what takes place in its battery. It does not matter where the vehicles are, or which trip they would be serving. In this system, a vehicle can only be active, idle, slow charging, or fast charging. To make the problem simpler, the operation day is divided into time-steps of 30 minutes.

The charging curve is a piecewise linear function with a lower charging rate above 80%. The idea for imposing a different charging rate above 80% SoC is to assume two imaginary batteries for each car, with capacities of 80 and 20. The second one only can charge when the first one is full, and the first one can only decharge if the second one is empty. Demand is given, the more demand the fleet can satisfy the better. Vehicles pay for charging and are penalized every time they start to charge. This is to prevent a plan with frequent and short charging periods. The price of charging is per KW gained, and relative to the price of overnight charging in the depot. In the real scenario the operator not only has to have enough charged cars but also have them scattered spatially; hence a parameter is introduced as the desired number of charged cars for each time-step. Satisfying the number of desired charged cars is rewarded in the optimization model.

3.2.2. Input

The basic inputs for the optimization model include fleet size, number of chargers, the initial charge of vehicles, and charging rate. The more complex input, which would have to be derived from the operation data, include the number of active vehicles per time-step, the number of charged cars required, and the average consumption rate per time-step (30 minutes). Furthermore, to determine weights in the objective function, the operator needs estimates of average fare from 30 minutes of a vehicle's operation, the average cost of going to a charging station, along with electricity prices through the day. Objective weights will be discussed further after the formulation of the problem is presented.

3.2.3. Mathematical formulation

Decision variables:(all variables are non-negative)

Independent variables:

$x\_start\_slow_{it}$	Is 1 if vehicle i starts slow charging, at t. It is 0 otherwise.
$x\_stop\_slow_{it}$	Is 1 if vehicle i stops slow charging, at t. It is 0 otherwise.
$x\_start\_fast_{it}$	Is 1 if vehicle i starts fast charging, at t. It is 0 otherwise.
$x\_stop\_fast_{it}$	Is 1 if vehicle i stops fast charging, at t. It is 0 otherwise.
d_{it}	Is 1 if vehicle i is serving demand at time t. It is 0 otherwise.

Dependent variables:

u_slow_{it}	The amount of charge that vehicle i would have above 80%, at t, if it had a constant charge rate while charging with slow charger. (Real number)
u_fast_{it}	The amount of charge that vehicle i would have above 80%, at t, if it had a constant charge rate while charging with fast charger. (Real number)
v_{it}	The amount of charge that vehicle i spend from the first 80% of the battery, at t. (Real number)
w_{it}	The amount of charge that exceeds 100 if vehicle i was in charge entire time-step, at t. (Real number)
a_slow_{it}	It is 1 if SoC of vehicle i went higher than 80%, at t, while charging with slow charger. It is 0 otherwise.
a_fast_{it}	It is 1 if SoC of vehicle i went higher than 80%, at t, while charging with a fast charger. It is 0 otherwise.
b_{it}	It is 1 if SoC of vehicle i went lower than 80%, at t, while doing a trip. It is 0 otherwise.
q_{it}	It is 1 if the SoC of vehicle i is higher than 20%, and it is not busy, at t. It is 0 otherwise.
Q_t	It is the slack for the desired number of vehicles with SoC higher than 20. (Real number)

Parameters:

(The codes (A*) are cross references to Table 4.2 where the values of these parameters are specified)

$ChargingRate\_slow_1$	Charging rate of slow charger up to SoC of 80% (fraction per time-step)
$ChargingRate\_slow_2$	Charging rate of slow charger after SoC of 80% (fraction per time-step)
$ChargingRate\_fast_1$	Charging rate of fast charger up to SoC of 80% (fraction per time-step)
$ChargingRate\_fast_2$	Charging rate of fast charger after SoC of 80% (fraction per time-step)
$DechargingRate$	Battery consumption (fraction per time-step)
$NumberOfChargers\_slow$	Number of slow chargers
$NumberOfChargers\_fast$	Number of fast chargers
E_slow_t	Penalty for cost of electricity, proportional to additional electricity price at t compared to overnight charging
E_fast_t	Penalty for cost of electricity, proportional to additional electricity price at t compared to overnight charging
D_t	Number of required active vehicles at t
B_t	Number of desired charged cars at t
h	Number of steps in an operational day
L_{i0}	Initial charge of vehicle i
$L\_1_{i0} = min(L_{i0}, 0.8)$	Initial charge of vehicle i in the first imaginary battery
$L\_2_{i0} = max(L_{i0} - 0.8, 0)$	Initial charge of vehicle i in the second imaginary battery
β	Negative weight for satisfying demand (A1)
γ	Penalty for slack of number of charged cars (A2)
α_slow	Penalty for a trip to a slow charging station (A3)
α_fast	Penalty for a trip to a fast charging station (A4)
$MaxVisitCharging$	Maximum allowed number of visits to charging station by any vehicle

Substituting variables for conciseness in presentation of constraints

The following are not decision variables (dependent or independent). They are terms to represent additive functions of variables already defined, since these particular functions repeatedly appear in the constraints. y_slow_{it} is 1 when vehicle i is slow charging at t, it is 0 otherwise:

$$y\_slow_{it} = \sum_{k=0}^{t} (x\_start\_slow_{ik} - x\_stop\_slow_{ik}) \tag{3.1}$$

y_fast_{it} is 1 when vehicle i is fast charging at t, it is 0 otherwise:

$$y\_fast_{it} = \sum_{k=0}^{t} (x\_start\_fast_{ik} - x\_stop\_fast_{ik}) \tag{3.2}$$

Equations 3.3 and 3.4 along with Figure 3.3 explain how charge for the first and second imaginary batteries are calculated. The gray is the existing battery, the green is the added charge in one time-step, according to charging rate of L_1 (the higher rate), in red is the amount of battery we want to consume in one time-step. The ratio of u' as specified in the figure is obtained by multiplying u by $\frac{charging\ rate\ from\ 80\%\ to\ 100\%}{charging\ rate\ from 0\%\ to\ 80\%}$ to account for the lower charging rate above 80%.

L_1_{it} is charge in the first imaginary battery (upto 80%), in vehicle i, at t:

$$\begin{aligned} L\_1_{it} = \sum_{k=0}^{t} (&ChargingRate\_slow_1 \times y\_slow_{ik} + ChargingRate\_fast_1 \times y\_fast_{ik} - \\ &u\_slow_{ik} - u\_fast_{ik} - v_{ik}) + L\_1_{i0} \end{aligned} \tag{3.3}$$

L_2_{it} is charge in the second imaginary battery (last 20%), in vehicle i, at t:

$$\begin{aligned} L\_2_{it} = \sum_{k=0}^{t} (&\frac{ChargingRate\_slow_2}{ChargingRate\_slow_1} \times u\_slow_{ik} + \frac{ChargingRate\_fast_2}{ChargingRate\_fast_1} \\ &\times u\_fast_{ik} - DechargingRate \times d_{ik} + v_{ik} - w_{ik}) + L\_2_{i0} \end{aligned} \tag{3.4}$$

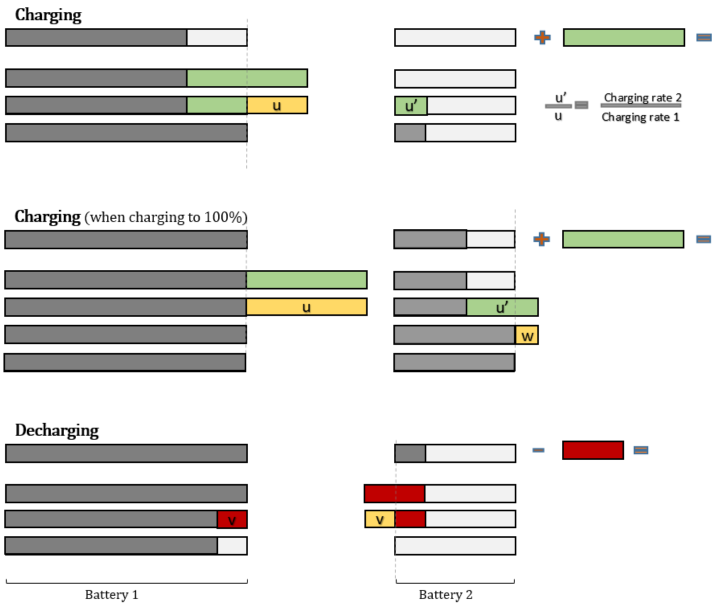


Figure 3.3: Charging and decharging in imaginary batteries

$L\_2\_shortage_{it}$ is shortage in the second imaginary battery (last 20%), in vehicle *i*, at *t*, for example, it is 0.05 if SoC is 85% and we need to consume 10% and is -0.05 if SoC is 95% and we need to consume 10%:

$$L\_2\_shortage_{it} = -(L\_2_{it-1} - DechargingRate \times d_{it}) \tag{3.5}$$

$L\_1\_plus\_(slow, fast)_{it}$ is the charge in the first battery in vehicle *i*, at *t-1* plus $ChargingRate\_(slow, fast)_1$ if the vehicle was in (slow, fast) charge at *t*:

$$L\_1\_plus\_slow_{it} = L\_1_{it-1} + ChargingRate\_slow_1 \times y\_slow_{it} \tag{3.6}$$

$$L\_1\_plus\_fast_{it} = L\_1_{it-1} + ChargingRate\_fast_1 \times y\_fast_{it} \tag{3.7}$$

Objective:
Minimize: cost of going to charge + charging cost - reward from active vehicles + cost of not having enough charged cars

$$\begin{aligned} Minimize: \alpha\_slow \sum_t \sum_i x\_start\_slow_{it} + \alpha\_fast \sum_t \sum_i x\_start\_fast_{it} + \\ \sum_t E\_slow_t (\sum_{i=0}^{n} ChargingRate\_slow_1 \times y\_slow_{it} - \sum_{i=0}^{n} u\_slow_{it} + \sum_{i=0}^{n} u\_slow_{it} \frac{ChargingRate\_slow_2}{ChargingRate\_slow_1}) + \\ \sum_t E\_fast_t (\sum_{i=0}^{n} ChargingRate\_fast_1 \times y\_fast_{it} - \sum_{i=0}^{n} u\_fast_{it} + \sum_{i=0}^{n} u\_fast_{it} \frac{ChargingRate\_fast_2}{ChargingRate\_fast_1}) + \\ \beta \sum_t \sum_i d_{it} + \gamma \sum_t Q_t \end{aligned} \tag{3.8}$$

Subject to:
On charger capacity: Total number of vehicles in charge is bounded by number of chargers owned by the operator:

$$\sum_i y\_slow_{it} \leq NumberOfChargers\_slow, \forall t \tag{3.9a}$$

$$\sum_i y\_fast_{it} \leq NumberOfChargers\_fast, \forall t \tag{3.9b}$$

On battery capacity: The total battery should stay between 0 and 1:

$$0 \leq L\_1_{it} \leq 0.8\,, \forall i, t \tag{3.10a}$$
$$0 \leq L\_2_{it} \leq 0.2\,, \forall i, t \tag{3.10b}$$
(3.10c)

No more trips than demanded: Number of vehicles active cannot be more than the demand:

$$\sum_i d_{it} \leq D_t\,, \forall t \tag{3.11}$$

In charge state is 0 or 1:

$$0 \leq y\_slow_{it} \leq 1\,, \forall i, t \tag{3.12a}$$
$$0 \leq y\_fast_{it} \leq 1\,, \forall i, t \tag{3.12b}$$
(3.12c)

If any vehicle is marked as active, it cannot go to charging at the same time:

$$d_{it} + q_{it} + y\_slow_{it} + y\_fast_{it} \leq 1\,, \forall i, t \tag{3.13}$$

On the number of charged vehicles desired: Number of charged vehicles plus the slack variable for number of charged vehicles required should be larger than the number of charged vehicles required.

$$\sum_i q_{it} + \sum_i d_{it} + Q_t \geq B_t\,, \forall t \tag{3.14}$$

$q_{it} = 0$ if SoC is less than 20%

$$q_{it} \leq L\_1_{it-1} + 0.8\,, \forall i, t \tag{3.15}$$

The relation between v_{it} and b_{it}: charge from the second assumed battery has to run out before using the first assumed battery. As a reminder $L\_2\_shortage_{it}$ is shortage in the second imaginary battery, $L\_2\_shortage_{it} = -(L\_2_{it-1} - DechargingRate \times d_{it})$. It is positive if we need to use the first battery (0 to 80 percent), it is negative if we have enough charge in the second battery:

The following constraint is active when $b_{it} = 1$, which means that we are using L_1. It is to bound v_{it} from above:

$$v_{it} \leq L\_2\_shortage_{it} + (1 - b_{it}) \times 0.8\,, \forall i, t \tag{3.16}$$

The following constraint is active when $b_{it} = 0$, which means that we are not using L_1. It is to bound v_{it} to 0:

$$v_{it} \leq b_{it} \times 0.8\,, \forall i, t \tag{3.17}$$

The following constraint sets $b_{it} = 0$, if charge in L_2 is higher than $DechargingRate$ (enough charge in the second battery):

$$b_{it} \leq 1 + L\_2\_shortage_{it}\,, \forall i, t \tag{3.18}$$

The following constraint sets $b_{it} = 0$, when $L\_2 = 0$ and $d_{it} = 0$:

$$b_{it} \leq d_{it}\,, \forall i, t \tag{3.19}$$

The relation between a_slow, a_fast and u_slow, u_fast: charge from the second assumed battery has to run out before using the first assumed battery. Written in form of general [x, y, u, a], same holds for subscripts "_slow" and "_fast". Equation 3.20 is the main one we need to set the value of u, when the charging exceeds the first battery onto the second battery, successive equations 3.21 to 3.23 will accommodate the first one.

The following constraint is active when $L\_1\_plus_{it} > 0.8$ and $a_{it} = 1$, which means that L_2 is getting charged. It will bound u_{it} from above:

$$u_{it} \le L\_1\_plus_{it} - 0.8 \times a_{it}, \forall i, t \tag{3.20}$$

The following constraint is active when $a_{it} = 0$, which means that L_2 is not getting charged. It will set $u_{it} = 0$:

$$u_{it} \le a_{it}, \forall i, t \tag{3.21}$$

The following constraint sets $a_{it} = 0$, if L_1 is not full:

$$a_{it} \le 0.2 + L\_1\_plus_{it}, \forall i, t \tag{3.22}$$

The following constraint sets $a_{it} = 1$, if L_1 is full:

$$L\_1\_plus_{it} - 0.8 \le a_{it}, \forall i, t \tag{3.23}$$

The following constraint sets $a_{it} = 0$, when L_1 is full but vehicle is not charging:

$$a_{it} \le y_{it}, \forall i, t \tag{3.24}$$

To limit the search space, as a heuristic, the maximum number of visits of each vehicle to chargers is restricted:

$$\sum_t x\_start\_slow_{it} + \sum_t x\_start\_fast_{it} \le MaxVisitCharging, \forall i \tag{3.25}$$

3.2.4. Solving with Column Generation

In the beginning of the day, where all vehicles have the same SoC, sets of decision variables associated with each vehicle are interchangeable with each other. Therefore with fleet size n, there are $n!$ potential solutions that are the same. Column Generation (CG) can take advantage of this symmetry between vehicles in the problem. The Dantzig-Wolfe decomposition is used to divide the original problem into a sub-problem that generates daily plans for individual vehicles, given their battery constraints, and a master problem that decides how many vehicles should follow each of the plans generated by the sub-problem. The reader can refer to [16] for more details on the Column Generation method. The master problem will make sure that global constraints (associated with more that one vehicle) such as limited charger capacity, available demand and the desired number of charged cars are satisfied.

CG sequentially solves the master and the pricing problem; at each iteration of the column generation heuristic, the dual multiplier obtained by solving the linear relaxation of the master problem are used to generate daily plans that will improve the objective of the relaxed master problem. The algorithm will terminate if there is no significant improvement over the last 15 iterations. It is trivial that the columns (daily plans from the sub-problem) generated in earlier stages have lower quality, and will stop appearing as basic variables in the master problem after a while. Therefore to trim the bad options, every column that has not been chosen for the past 15 iterations will be eliminated from the set. After the last iteration, the master problem will find an integer solution, given the restricted set of columns already generated. The resulting solution is not optimal and the gap cannot be quantified using the Column Generation technique. The reason is twofold, one because the new columns (individual car plans) are generated based on the, relaxed master problem, and columns that may improve the integer master problem are not known; two because the iteration is terminated once improvement of the objective is slowed down, and not when no new column is available. One may still compere the objective of the Column Generation solution with the bound of the original problem, however obtaining a good bound on the complete problem with more that 100 vehicles is challenging.

The initial columns for this problem are generated by solving an aggregated version of the original problem sub-optimally. For example for a fleet of size 400, the vehicles were clustered in groups of 40; where within each group vehicles will act identical to each other. In this example, we would have 10 daily plans for individual vehicles, to start the Column Generation with. Algorithm 1 describes the iterative procedure.

Algorithm 1 Column Generation

Initialize a set of daily plans (columns) for individual cars by solving the original problem
while True **do**
  Solve the linear relaxation of the master problem
  if If improvement in the objective of relaxed master problem over past 15 iterations is less than 0.5% **then**
    terminate loop
  Get dual multipliers form relaxed master problem
  Calculate the dual multipliers $\pi\_x\_start\_slow_t$, $\pi\_x\_stop\_slow_t$, $\pi\_x\_start\_fast_t$, and $\pi\_x\_stop\_fast_t$ with Equations 3.26 to 3.29
  Solve the sub-problem, generating new daily plans (columns)
  Insert the corresponding columns in the master problem
  if any column not used for the past 15 iteration **then**
    Eliminate column
Solve the master problem

CG formulation

Pricing sub-problem

Decision variables:

Independent variables:

$x\_start\_slow_t$	Is 1 if vehicle starts slow charging, at t. It is 0 otherwise.
$x\_stop\_slow_t$	Is 1 if vehicle stops slow charging, at t. It is 0 otherwise.
$x\_start\_fast_t$	Is 1 if vehicle starts fast charging, at t. It is 0 otherwise.
$x\_stop\_fast_t$	Is 1 if vehicle stops fast charging, at t. It is 0 otherwise.
d_t	Is 1 if vehicle is serving demand at time t. It is 0 otherwise.

Dependent variables (for description refer to original problem A):

u_slow_t	u_fast_t	v_t
w_t	a_slow_t	a_fast_t
b_t	q_t	

Parameters:

The following are dual multiplier from the relaxed master problem:

$\pi\_y\_slow_t$	$\pi\_y\_fast_t$
π_d_t	π_q_t

Where dual multiplier for y can be translated to dual multipliers for x_start and x_stop with following equations. Basically for each x_start and x_stop the dual multiplier is calculated by summing over the dual multiplier of y variables that they influence. In equation 3.26 for example, $x\_start\_slow_t$ has weight of 1 in all y_slow_k variables, where k is from t until end of the day.

$$\pi\_x\_start\_slow_t = \sum_{k=t}^{h} \pi\_y\_slow_k \tag{3.26}$$

$$\pi\_x\_stop\_slow_t = -\sum_{k=t}^{h} \pi\_y\_slow_k \tag{3.27}$$

$$\pi\_x\_start\_fast_t = \sum_{k=t}^{h} \pi\_y\_fast_k \tag{3.28}$$

$$\pi\_x\_stop\_fast_t = -\sum_{k=t}^{h} \pi\_y\_fast_k \tag{3.29}$$

Refer to original problem A for description of the following parameters:

$ChargingRate\_slow_1$	$ChargingRate\_slow_2$	$ChargingRate\_fast_1$
$ChargingRate\_fast_2$	$DechargingRate$	E_t
h	L_1_0	L_2_0
α_slow	α_fast	$MaxVisitCharging$

Substituting variables for conciseness

$$y\_slow_t = \sum_{k=0}^{t} (x\_start\_slow_k - x\_stop\_slow_k) \tag{3.30}$$

$$y\_fast_t = \sum_{k=0}^{t} (x\_start\_fast_k - x\_stop\_fast_k) \tag{3.31}$$

$$\begin{split} L\_1_t = \sum_{k=0}^{t} (ChargingRate\_slow_1 \times y\_slow_k + \sum_{k=0}^{t} ChargingRate\_fast_1 \times y\_fast_k - \\ u\_slow_k - u\_fast_k - v_k) + L\_1_0 \end{split} \tag{3.32}$$

$$\begin{split} L\_2_t = \sum_{k=0}^{t} (\frac{ChargingRate\_slow_2}{ChargingRate\_slow_1} \times u\_slow_k + \frac{ChargingRate\_fast_2}{ChargingRate\_fast_1} \\ \times u\_fast_k + v_k + L_0 - DechargingRated_k - w_k) + L\_2_0 \end{split} \tag{3.33}$$

$$L\_2\_shortage_t = -(L\_2_{t-1} - DechargingRate \times d_t) \tag{3.34}$$

$$L\_1\_plus\_(slow, fast)_t = L\_1_{t-1} + ChargingRate\_(slow, fast)_1 \times y\_(slow, fast)_t \tag{3.35}$$

Objective:

$$\begin{split} \alpha\_slow \sum_t x\_start\_slow_t + \alpha\_fast \sum_t x\_start\_fast_t + \\ \sum_t E\_slow_t (ChargingRate\_slow_1 \times y\_slow_t - u\_slow_t + u\_slow_{it} \frac{ChargingRate\_slow_2}{ChargingRate\_slow_1}) + \\ \sum_t E\_fast_t (ChargingRate\_fast_1 \times y\_fast_t - u\_fast_t + u\_fast_t \frac{ChargingRate\_fast_2}{ChargingRate\_fast_1}) + \\ \beta \sum_t d_t + \\ - \sum_t d_t \times \pi\_d_t - \sum_t x\_start\_slow_t \times \pi\_x\_start\_slow_t - \sum_t x\_stop\_slow_t \times \pi\_x\_stop\_slow_t \\ - \sum_t x\_start\_fast_t \times \pi\_x\_start\_fast_t - \sum_t x\_stop\_fast_t \times \pi\_x\_stop\_fast_t + \sum_t q_t \times \pi\_q_t \end{split} \tag{3.36}$$

The following is the cost of each plan (calculated after solving the problem) which will be passed on to the master problem:

$$\begin{split} C = \alpha\_slow \sum_t x\_start\_slow_t + \alpha\_fast \sum_t x\_start\_fast_t + \\ \sum_t E\_slow_t (ChargingRate\_slow_1 \times y\_slow_t - u\_slow_t + u\_slow_{it} \frac{ChargingRate\_slow_2}{ChargingRate\_slow_1}) + \\ \sum_t E\_fast_t (ChargingRate\_fast_1 \times y\_fast_t - u\_fast_t + u\_fast_t \frac{ChargingRate\_fast_2}{ChargingRate\_fast_1}) + \\ \beta \sum_t d_t \end{split} \tag{3.37}$$

Subject to:
On battery capacity: Battery should stay between 0 and 100

$$0 \le L\_1_t \le 0.8, \forall t \tag{3.38a}$$
$$0 \le L\_2_t \le 0.2, \forall t \tag{3.38b}$$
$$\tag{3.38c}$$

y_t is 0 or 1

$$0 \leq y\_slow_t \leq 1, \forall t \tag{3.39a}$$
$$0 \leq y\_fast_t \leq 1, \forall t \tag{3.39b}$$

If any vehicle is marked as active, it cannot go to charging at the same time

$$d_t + q_t + y\_slow_t + y\_fast_t \leq 1, \forall t \tag{3.40}$$

$$q_t \leq L\_1_t + 0.8, \forall t \tag{3.41}$$

The relation between v_t and b_t:

$$v_t \leq L\_2\_shortage_t + (1 - b_t) \times 0.8, \forall t \tag{3.42}$$

$$v_t \leq b_t \times 0.8, \forall t \tag{3.43}$$

$$L\_2\_shortage_t \leq b_t, \forall t \tag{3.44}$$

$$b_t \leq d_t, \forall t \tag{3.45}$$

The relation between a_slow, a_fast and u_slow, u_fast: Written in form of general [x, y, u, a], same holds for subscripts "_slow" and "_fast"

$$u_t \leq L\_1\_plus_t - 0.8 \times a_t \; \forall t, \forall t \tag{3.46}$$

$$u_t \leq a_t, \forall t \tag{3.47}$$

$$a_t \leq 0.2 + L\_1\_plus_t, \forall t \tag{3.48}$$

$$L\_1\_plus_t - 0.8 \leq a_t, \forall t \tag{3.49}$$

$$a_t \leq y_t, \forall t \tag{3.50}$$

$$\sum_t x\_start\_slow_t + \sum_t x\_start\_fast_t \leq MaxVisitCharging \tag{3.51}$$

Master problem
Decision variables:
Independent variables:

k_i — Number of vehicles that follow plan i. (integer)

Dependent variables:

Q_t — It is the slack for the desired number of vehicles with SoC higher than 20. (Real number)

Parameters:

C_i — Cost of plan i. Where C_i is the objective of sub-problem excluding the dual multipliers
y_slow_{it} — Is 1 if vehicle is in charge in plan i, at t
y_fast_{it} — Is 1 if vehicle is in charge in plan i, at t
d_{it} — Is 1 if vehicle is serving demand in plan i, at t
q_{it} — Is 1 if vehicle has SoC higher than 20% and is not busy in plan i, at t

Refer to original problem A for description of the following parameters:

$NumberOfChargers\_slow$ $NumberOfChargers\_fast$ D_t
B_t h β
γ

Objective:

$$Minimize: \sum_i C_i k_i + \gamma \sum_t Q_t \tag{3.52}$$

Subjected to:
On charger capacity: Total number of vehicles in charge is bounded by number of chargers owned by the operator

$$\sum_i k_i \times y\_slow_{it} \leq NumberOfChargers\_slow, \forall t \tag{3.53a}$$

$$\sum_i k_i \times y\_fast_{it} \leq NumberOfChargers\_fast, \forall t \tag{3.53b}$$

No more trips than demanded: Number of vehicles active cannot be more than the demand

$$\sum_i k_i d_{it} \leq D_t, \forall t \tag{3.54}$$

On the number of charged vehicles desired

$$\sum_i k_i q_{it} + \sum_i k_i d_{it} + Q_t \geq B_t, \forall t \tag{3.55}$$

3.2.5. Final comments on daily planning

The following considerations are discussed in this section:

1. Problem with lack of spatial dimension
2. Challenge in providing parameters required
3. Overestimation vs underestimation of demand

Planning with limited information does not come without cost. Certain aspects of the problem cannot be picked up by the model. Lacking spatial resolution, algorithm $\underline{A}$ can inherently leverage a flexibility in choosing vehicles for demand and sending to charge that in reality does not exist, or it has higher than average cost. For example cost of traveling to some charging stations (the ones further away from typical pathways of vehicles) is higher. Therefore using the full capacity of all charging stations is more expensive than using 80% of them. This is adjustable by introducing an extra penalty on using the last set of chargers. Other flexibilities that might result in unfeasible daily charging solutions are the ability of algorithm $\underline{A}$ to choose vehicles with lowest SoC for charging, and keep the highest SoC's active, or any other specific strategy based on SoC that the operator is not promoting in operation.

As stated previously, the objective of optimization includes a penalty if the number of desired charge vehicles is not met. To recap, the number of charged vehicles desired is a way of communicating constraints imposed by the spatial dimension of vehicles. If the operator has to serve 20 passengers, having only 20 charged vehicles is not enough, the vehicles should be scattered through the operating zone. However, the operator needs a function to map the number of vehicles that will be active, to the number of vehicles that have to have the capacity to be active. Once obtained this function is assumed to be fixed for all operation days. Further values to be objectified by the operator are weights of the objective function. The weights for going to charge, charging, and losing a customer can be roughly calibrated by having the average charge spent for sending a vehicle to charging station, and the ratio of the operators profit to energy price for average trips. The weight for the cost of not having enough charged cars is harder to calibrate.

The last issue is that the operator never has one estimation of demand or the number of active vehicles. It is not part of the scope to deal with stochastic or multi-scenario demand. However, one can consider updating the single input of active vehicles that are given. The costs of underestimating, and overestimating demand are not symmetric. If the operator overestimates demand, it will result in lower efficiency per passenger, by visiting charging stations more frequently than needed, charging in high electricity price times, and by having fewer vehicles available for operation, which increases the routing cost. If the operator underestimates demand, it will result in rejecting customers.

It is impossible to quantify the extra cost imposed by overestimating demand by 10%, or the loss of revenue by underestimating demand by 10%, via the current model. They can be subjectively tested in the case study. However, we can already expect that the impact of overestimating grows faster than linear, meaning that the first 5% may not impose any cost at all, but a 20% overestimation may become very costly. Therefore it is useful to look at the changes in efficiency per passenger with changes in total demand.

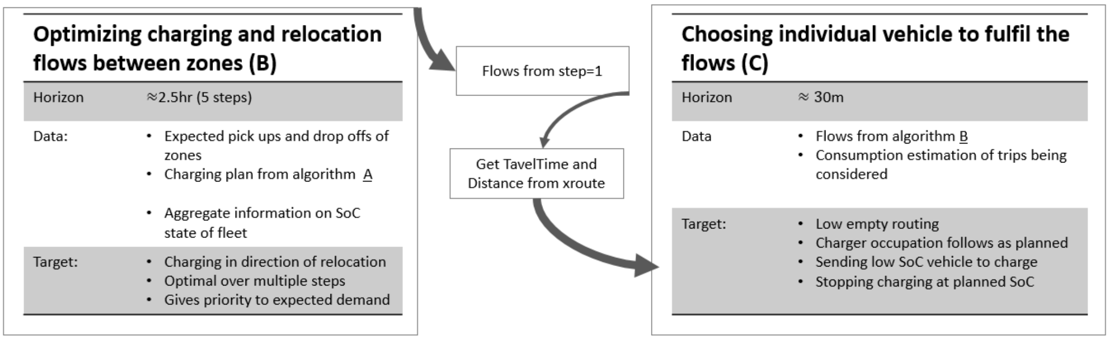


Figure 3.4: Summary of online planing

3.3. Online charging planner

3.3.1. Outline

As previously explained, the online planner needs an aggregated level optimization to take advantage of aggregated pickup and drop-off information over a two hour horizon. The aggregated decisions are empty vehicle flows between zones over which data on pickup an drop-offs are available. Next to the operating zones, there are charger zones, each including a bundle of charging stations close together. The charger zones do not need to be within boundaries of normal zones. The output of aggregated level optimization (algorithm B) is then the number of vehicles moving between operating zone, going from operating zone to charger zones, and from charger zones to operating zones.

After obtaining results of algorithm B, the operator knows that it is best, for example, to re-position k vehicles from zone i to zone j and send q vehicles from zone i to charger zone m, within the next 30 minutes. Knowing the time and location of the vehicles' last drop-offs, the operator can choose the best vehicles to fulfill the recommended flows by the algorithm B. This is a job for the optimization algorithm C. Algorithm C will try to minimize the routing cost of going to charge and relocation trips. It makes sure charger utilization will follow as planned, it chooses the least charged vehicles for going to charge, and that in-charge vehicles stop charging at the planned SoC (from algorithm A). A summary of online planning is presented in Figure 3.4.

Time to solve algorithm B and C are interdependent, and rely on the size (and consequently number) of the zones. The larger the size of the zone (the smaller the number) algorithm B becomes less computationally costly in expense of a longer preparation and solve time for algorithm C. Algorithm C has a higher cost precision than algorithm B, meaning that its estimate of routing cost is better. In addition, it knows the SoC of the individual vehicles, and it can therefore, choose the lowest charged vehicles to send to charge, and the high charged vehicles for relocation. With that said, we can see that cost of charging trips can be reduced with making the zones larger (allocating greater flexibility to algorithm C). However, size of zone will also affect the quality/resolution of pickup estimation. For the purpose of relocation, average number of vehicles per zone should not be less than five (lower bound on size of the zones). Furthermore, as a rule of thumb, 80% percentile of travel time within a zone should not exceed the maximum waiting time of passengers, as we expect the pickups form a zone to be satisfied by the vehicles in that zone (upper bound on size of the zones).

3.3.2. Algorithm B

Input

The first input of the algorithm is the initial location of the vehicles. In algorithm B the output of the first time-step is interpreted as the vehicles that should start a charging or relocation trip during the next 30 minutes from one zone to the other. Therefore, all vehicles with their Expected arrival time (EAT) earlier than 30 minutes count as an initial number of vehicles in their destination zone. If a vehicle has a planned route ending between 30 and 60 minutes, it is counted as a drop-off at the beginning of the second time-step, and therefore usable in the second time-step of algorithm B. Since the flows from the first step of algorithm B will be enforced in algorithm C, they have to be feasible. Meaning not only the operator has to have vehicles in the zone to be able to relocate them, but they have to have enough charge to do so. In the same way, there can be no flows from zone i to charger zones if all vehicles in zone i are 90% charged. That is why for the first time-step extra constraints are enforced. This limitation is relaxed for next time-steps since we no longer have information on the distribution of charged vehicles among the zones. Relying on the total number of vehicles available at each zone is the best the operator can do.

Algorithm B gets the number of vehicles that have to be in charge in each time-step, along with the least number of vehicles who have to go to charge. The latter is required to distribute charge among vehicles, otherwise algorithm B would just keep the same vehicles in charge, to avoid the cost of routing vehicles in and out of chargers. While algorithm B will respect the amount of in charge vehicles recommended by A, however, the number of vehicles going to charge is interpreted as a lower bound. This way algorithm B can cut charging of vehicles in zones where vehicles are needed and instead add extra a charging trip in another zone, to avoid an expensive relocation. While these two constraints apply on the union of all chargers, for each charger zone, we yet need to enforce a minimum and a maximum number of vehicles who can stop charging. Maximum is required to respect the battery capacity of vehicles. Otherwise, algorithm B can assign high turnover to central chargers, while keeping some vehicles fixed in chargers further away. Minimum is required, not to cut charging of vehicles that have just arrived at charging.

The last inputs required, are the number of pickup and drop-offs per zone over each step over the horizon, and the average cost for going from one zone to the other. While calculating the number of pickups and drop-offs operator should account for the rate of shared vehicles. Imagine if two passengers have the same origin zone, and will share their ride, then the operator certainly does not need to relocate two vehicles to the zone to serve those customers.

Mathematical formulation

Parameters:

(The codes (B*) are cross references to Table 4.2 where the values of these parameters are specified)

$i, j \in$ Set of zones

$n, m \in$ Set of charging station zones

h	Horizon of stage B
v_{i1}	Number of cars in zone i at t=1
v_{n1}	Number of cars in charging station zone n at t=1
$v_i^{chargeable}$	Number of cars in zones i that have lower SoC than a $lower\_bound\_soc\_for\_going\_to\_charge_t$ = $mean\_SoC\_of\_vehicles\_going\_to\_charge\_in\_algorithm\_A_t + 0.15$ at time $t = 1$
$v_i^{available}$	Number of cars in zone i that have enough charge for at least one trip at time $t = 1$
$v_{nt}^{MustCutCharge}$	Number of cars in charging zone n that were put to charge before $t = 1$ and will reach $upper\_bound\_soc\_for\_being\_in\_charge_t$ = $min(mean\_SoC\_of\_vehicles\_stopping\_charge\_in\_algorithm\_A_t + 0.25, 100)$, at time t
$v_{nt}^{CanCutCharge}$	Number of cars in charging zone n that were put to charge before $t = 1$ and will reach $lower\_bound\_soc\_for\_stopping\_charge_t$ = $max(mean\_SoC\_of\_vehicles\_stopping\_charge\_in\_algorithm\_A_t - 0.25, 35)$, at time t
$Charger\_capacity_n$	Charger capacity of charging station zone n
MCD_s	Minimum charging duration for slow charging
MCD_f	Minimum charging duration for fast charging
p_{it}	Number of pickups from zone i at time t
d_{jt}	Number of drop-offs from zone j at time t
C_{im}	Cost to go from zone i to zone m (in percent charge spent)
C_{ij}	Cost to go from zone i to zone j (in percent charge spent)
C_{nj}	Cost to go from zone n to zone j (in percent charge spent)

Y_slow_t Number of vehicles that should be in slow charge at t, from algorithm A
X_slow_t Number of vehicles that should go to slow charge at t, from algorithm A
Y_fast_t Number of vehicles that should be in fast charge at t, from algorithm A
X_fast_t Number of vehicles that should go to fast charge at t, from algorithm A
α Penalty for unsatisfied pickup, it is an upper bound to charge spent on a relocation trip (B1)
β_slow Penalty for unsatisfied slow in-charge (B2)
β_fast Penalty for unsatisfied fast in-charge (B3)
θ_slow Penalty for unsatisfied going to slow charge (B4)
θ_fast Penalty for unsatisfied going to fast charge (B5)
β' Penalty for surplus in-charge (B6)

Decision variables: (all variables are non-negative)

x_{ijt} Empty vehicle flow from i to j (relocation trip) at t
x_{imt} Empty vehicle flow from i to m (charging trip) at t
x_{njt} Empty vehicle flow from n to j (relocation from the chargers) at t
s_{it} Slack for pickups of zone i at t
B_slow_t Slack for number of vehicles in slow charge at t
B_fast_t Slack for number of vehicles in fast charge at t
G_slow_t Slack for number of vehicles going to slow charge at t
G_fast_t Slack for number of vehicles going to fast charge at t
B'_slow_t Surplus of number of vehicles in slow charge at t
B'_fast_t Surplus of number of vehicles in fast charge at t

Notation shortcuts:

$$S_t = \sum s_{it} \tag{3.56}$$

$$P_t = \sum p_{it} \tag{3.57}$$

Objective:

The objective is to minimize sum over all routing costs in addition to the penalty for unsatisfied pickups and in charge vehicles.

$$\begin{aligned} Minimize \sum_{t=1}^{h} \Bigg(& \sum_{i,j} C_{ij} x_{ij,t} + \sum_{i,m} C_{im} x_{imt} + \sum_{n,j} C_{nj} x_{njt} + \alpha S_t + \\ & \beta\_slow \times B\_slow_t + \beta\_fast \times B\_fast_t + \\ & \theta\_slow \times G\_slow_t + \theta\_fast \times G\_fast_t + \\ & \beta' \times B'\_slow_t + \beta' \times B'\_fast_t \Bigg) \end{aligned} \tag{3.58}$$

Subject to:

Number of vehicles in zone i should be positive at time t, when all vehicle with origin i have left from the zone, but no vehicle with destination i has arrived into the zone. Therefore, it is calculated as initial number of vehicles minus number of vehicles who left the zone from beginning to t, plus number of vehicles that have arrived to zone i before t. Furthermore, number of drop offs predicted for the zones is discounted by $\left(1 - \frac{S_k}{P_k}\right)$, that is rate of satisfies pickups. It means that in each time-step number of total pickups and total drop-offs should be the same. This prevents double counting of vehicles which is caused by slack variable of pickups.

$$\begin{aligned} v_{i1} + \sum_{k=1}^{t} \left(-(p_{ik} - s_{ik}) - \sum_{i} x_{ijk} - \sum_{i} x_{imk} \right) + \sum_{k=1}^{t-1} \Bigg(d_{ik} \times \left(1 - \frac{S_k}{P_k}\right) + \\ \sum_{i} x_{jik} + \sum_{i} x_{nik} \Bigg) \geq 0, \forall t \in \{1, 2, \ldots h\}, i \end{aligned} \tag{3.59}$$

Number of vehicles in charging station zone n should be positive and lower than charger capacity of the zone at time t.

$$0 \leq v_{n1} + \sum_{k=1}^{t} \left(-\sum_{j} x_{njk} \right) + \sum_{k=1}^{t} \sum_{i} x_{ink} \leq Charger\_capacity_n \,, \forall t \in \{1,2,\dots h\}, n \tag{3.60}$$

The following constraint will prevent sending vehicles with high SoC e.g. 90% back to charge, for the first time-step (which will be enforced to algorithm C).

$$\sum_{m} x_{im1} \leq v_i^{chargeable} \,, \forall i \tag{3.61}$$

The following constraint will prevent relocating vehicles that do not have enough charge to serve any trip, for the first time-step. (The first time-step is the only one, which flows will be passed on to algorithm C)

$$\sum_{j} x_{ij1} \leq v_i^{available} - p_{i1} + s_{i1} \,, \forall i \tag{3.62}$$

The following constraints control that the number of vehicles in charge follow the plan by algorithm A. The surplus is to deal with the fact that vehicles actually arrive later in charging than B assumes, (B assumes they arrive immediately) and must sometimes stay later to achieve a certain SoC; hence sometimes number of in charge vehicles are allowed to be higher than planned by A. Same logic will also help in case we have underestimated demand, and vehicles than arrive into chargers have lower SoC than originally planned.

For the following equitation: $n, m \in$ Set of slow charging station zones

$$\sum_{n} v_{n1} + \sum_{k=1}^{t} \sum_{i} \sum_{m} x_{imk} - \sum_{k=1}^{t} \sum_{n} \sum_{j} x_{njk} = Y\_slow_{t+1} + B'\_slow_t \,, \forall t \in \{1,2,\dots h\} \tag{3.63}$$

For the following equitation: $n, m \in$ Set of fast charging station zones

$$\sum_{n} v_{n1} + \sum_{k=1}^{t} \sum_{i} \sum_{m} x_{imk} - \sum_{k=1}^{t} \sum_{n} \sum_{j} x_{njk} = Y\_fast_{t+1} - B\_fast_t + B'\_fast_t \,, \forall t \in \{1,2,\dots h\} \tag{3.64}$$

The following constraints control that the number of vehicles going to charge follow the plan by algorithm A. Without these constraints and just relying on number of in charge vehicles algorithm B will keep the same vehicles in charge, in other words, these constraints will bound the average duration of charge from above. Duration of charge is not bounded from below, since we can always cut charging of a vehicle as long we replace it with another one, if that reduces the routing cost.

For the following equitation: $n, m \in$ Set of slow charging station zones

$$\sum_{i} \sum_{m} x_{imt} + G\_slow_{t+1} \geq X\_slow_t \,, \forall t \in \{1,2,\dots h\} \tag{3.65}$$

For the following equitation: $n, m \in$ Set of fast charging station zones

$$\sum_{i} \sum_{m} x_{imt} + G\_fast_{t+1} \geq X\_fast_t \,, \forall t \in \{1,2,\dots h\} \tag{3.66}$$

The following controls that charged vehicle are not kept in any charging station zone. While the overall average duration of charge is bounded, we still need to constrain minimum number of outbound trips from individual charging station zone. Otherwise solution of algorithm B can include charging vehicles over 100%.

$$\sum_{k=1}^{t} \sum_{j} x_{njk} \geq \sum_{k=1}^{t} v_{nk}^{MustCutCharge} \,, \forall t \in \{1,2,\dots h\} \tag{3.67}$$

The following controls that vehicles have reached a Lower Bound SoC when they stop charging. This is crucial for $t = 1$ as we do not want to force a vehicle that just arrived to charging station to leave; it is important for $t : 2 - h$, to prevent algorithm B to plan short charging duration for central charging station, and long duration for stations further away.

$\forall n \in$ Set of slow charging station zones.

$$\sum_{k=1}^{t}\sum_{j} x_{njk} \leq \sum_{k=1}^{t} v_{nk}^{CanCutCharge} + \sum_{i}\sum_{k=1}^{t-MCD\_s} x_{ink}, \forall t \in \{1,2,\dots h\}, n \tag{3.68}$$

$\forall n \in$ Set of fast charging station zones:

$$\sum_{k=1}^{t}\sum_{j} x_{njk} \leq \sum_{k=1}^{t} v_{nk}^{CanCutCharge} + \sum_{i}\sum_{k=1}^{t-MCD\_f} x_{ink}, \forall t \in \{1,2,\dots h\}, n \tag{3.69}$$

The pickup slack is bounded by the number of pick ups. Without this constraint algorithm B can produce vehicles that do not exist, and use them to send to charge.

$$s_{it} \leq p_{it}, \forall t \in \{1,2,\dots h\}, i \tag{3.70}$$

Final comments on algorithm B

The following considerations are discussed in this section:

1. Limitations imposed by discretizing time
2. Coping with stochasticity in pickup information
3. Coping with lack of SoC information

One limitation of algorithm B, is that independent of the distance between the zones, each trip is assumed to take one full step. In view of algorithm B if a vehicle is being relocated at step t it cannot be used until step $t+1$. In reality, if travel time is well below 30 minutes, we can relocate and use the vehicle in the time equivalent to one step. One can adjust this by identifying the zones that do need cars, setting their flows to other operating zones to zero, and then allowing them to use cars from the adjutant zone. If outbound flows to other zones are not set to zero, the algorithm can exploit the zero travel time between nearby zones, to get anywhere within the same step. This correction is not implemented, and is only a suggestion.

In the current version of algorithm B, all relocation trips have the same weight. A more realistic alternative is to have a higher incentive for example for the first 80% percent of desired pickups, if there is more confidence that they will be realized. The same weighting mechanism can be exploited, if the operator has some information, and will prioritize based on the pooling potential of trips (e.g. trips from a set of origin zones are more likely to be shared, and thus more likely to be more profitable). If the information only refers to the origin zone, satisfying pickups of that zone will get more weight in the objective function. If there is information on pooling possibility based on origin and destination pair, and the operator is willing to reject the passengers whose ride have a low chance of pooling, then for each zone the percentage of the trips who can be shared should get higher value in algorithm B.

Further downside of accounting pickup numbers to be deterministic, is that algorithm will transfer just enough vehicles to the zone that presumably needs vehicles. However it can not optimize the likelihood of successful pickup by the excess vehicles at the time, if any. To make it tangible, imagine 2 zones A and B, A has 20 vehicles and needs 2, B has 2 vehicles but needs 10, current algorithm will only transfer 8 vehicles from A to B, without appreciating that the vehicles idle in zone A could have been more profitable, had they been moved to zone B, because in fact the 10 required vehicles is only an average. In the same way, if we have an extra vehicle, with two possible zones to relocate it to, A with 10 pickups, and B with 1, algorithm B will choose B if distance to B is lower. The remedy as mentioned before is having different types of pickups per zone, each having a reward reflecting the confidence in their realization. This way value of an extra relocation will decrease gradually, instead of going immediately to zero.

As stated previously algorithm B is based on the assumption that the operator does not have prior knowledge on the distribution of SoC vehicles after the first step. This means that algorithm B works well if there is good circulation between the zones, and the location of charged vehicles (except for the ones that are in charge) is almost independent of the location of chargers. In large operating areas where trips within two or more mega zones are much higher than the ones between them, the operator should either solve algorithm B separately for each mega zone, or somehow communicate the distribution of charged vehicles for the later time-steps.

3.3.3. Algorithm C

What algorithm C needs to achieve in terms of charging, is choosing the low charged vehicles who are close to charging stations, and reaching the targeted charger utilization (determined by the algorithm B) throughout the next 30 minutes. In terms of re-positioning, algorithm C should select charged vehicles that are the closest to the center of the destination zone. For going to charge and relocating, trip is assumed to take place after the last drop-off. For stopping to charge however, there is a time factor needed; thus to decide when in the 30 minutes each vehicle should stop charging, a time-step of 7.5 minutes is defined.

Input

First, potential vehicles, that can satisfy flows obtained from algorithm B should be specified. Being potential means that the vehicle has its last drop-off at the latest 30 minutes from now. If the destination is a charging station then the vehicle should be chargeable (have SoC lower than a threshold derived from A). If the destination is another zone, the vehicle should have an adequate charge for a relocation trip, plus an average trip with passengers. Then, the consumption of all vehicle-destination pairs will be calculated. If SoC of the vehicle is lower than the consumption, the option would be eliminated. With that said, there is always a reachable charger available for all vehicles (this is guaranteed by algorithm R), the elimination based on SoC might only filter chargers that are further away. For each pair, SoC of the vehicle at the last drop-off, EAT at potential destination point, and estimated consumption to the potential destination point is passed on as input.

Further inputs include the current state of chargers, along with EAT of vehicles on their way to charging stations[3]. Naturally, the flows, and the total number of in charge vehicles are passed down from the algorithm B. The last ingredients are the weights for the objective function, to calibrate the policies followed by algorithm C.

Mathematical formulation

Decision variables:

$n \in$ Set of chargers

$N \in$ Set of charging station zones

$I, J \in$ Set of zones

x_{iJ}	Vehicle i going to zone J after its last drop-off
x_{in}	Vehicle i going to charger n after its last drop-off
x_{iJt}	Vehicle i that is in charge, moves to zone J at time-step t
$s\_slow\_charge\_flow_{IN}$	Slack for charging flows from zone I to slow charging station zone N
$s\_fast\_charge\_flow_{IN}$	Slack for charging flows from zone I to fast charging station zone N
$s\_relocation\_flow_{IJ}$	Slack for relocation flows from zone I to zone J
$s\_in\_slow\_charge_t$	Slack for total in slow charge vehicles at time-step t
$s\_in\_fast\_charge_t$	Slack for total in fast charge vehicles at time-step t
$s'\_in\_slow\_charge_t$	Surplus for total in slow charge vehicles at time-step t
$s'\_in\_fast\_charge_t$	Surplus for total in fast charge vehicles at time-step t

Parameters:

(The codes (C*) are cross references to Table 4.2 where the values of these parameters are specified)

c_{iJ}	Charge consumption associated with vehicle i going to zone J
$c\_relocate\_soc_i$	Cost associated with SoC of vehicle i going to relocation
c_{in}	Charge consumption associated with vehicle i going to charger n
$c\_charge\_soc_i$	Cost associated with SoC of vehicle i going to charge
a_{int}	If the vehicle will arrive at charger n before time-step t
$c\_stopcharge\_soc_{it}$	Cost associated with SoC of vehicle i stopping to charge at time t
$B\_charge\_flow_{IN}$	Charging flow from zone I to N, from algorithm B
$c\_slow\_charge\_flow$	Penalty for slack of slow charging flows (C6)
$c\_fast\_charge\_flow$	Penalty for slack of fast charging flows (C8)
$B\_relocation\_flow_{IJ}$	Relocation flow from I to J, by algorithm B
$c\_relocation\_flow$	Penalty for slack of relocation flows (C4)
$B\_in\_slow\_charge$	Total in charge slow, by algorithm B
$B\_in\_fast\_charge$	Total in charge fast, by algorithm B

[3]Charging trips that were previously decided on but they have not yet arrived

$p\_in\_charge_{nt}$	Number of planned trips to charger n that arrive before t
$p\_in\_slow\_charge_t$	Number of planned trips to slow chargers that arrive before t
$p\_in\_fast\_charge_t$	Number of planned trips to fast chargers that arrive before t
$c\_in\_slow\_charge$	Penalty for slack of total in charge slow (penalty is for every 7.5 minutes (one-forth the step of algorithm B) that the charger is not used) (C5)
$c\_in\_fast\_charge$	Penalty for slack of total in charge fast (C7)
c'_in_charge	Penalty for surplus of total in charge (C9)
$B\_from\_charger\_flow_{NJ}$	Flow from charging station zone N to zone J, by algorithm B
$capacity_n$	Capacity of charging station n
$initial\_in\_charge_n$	Number of vehicles initially in charge at charging station n
$initial\_in\_slow\_charge$	Number of vehicles initially in slow charge
$initial\_in\_fast\_charge$	Number of vehicles initially in fast charge

where:

$B\_charge\_flow_{IN}$ Charging flow from zone I to N, from algorithm B is multiplied by $\frac{average\ expected\ start\ charging\ trip\ time\ of\ the\ vehicles\ candidate\ for\ going\ to\ charge}{0.5\times timestep\ duration\ of\ algorithm B}$, otherwise we force a flow that was indented over entire step of algorithm B, over a much shorter duration (e.g. 5 or 10 minutes), and grossly overestimate number of vehicles that should go to charge

$c\_relocate\_soc_i$ is negatively proportional to metric of the SoC group that the vehicle falls under, the metric is explained in Section 3.4.2: $C2 \times metric\_function(SoC_i)$

$c\_charge\_soc_i$ is proportional to SoC of vehicle: $C1 \times SoC_i$

$c\_stopcharge\_soc_{it}$ is proportional to rank of the action "stop charging of vehicle i at t that is in charger n" among all possible stopping charge actions for vehicles that are in charger n. The ranking rule is as follows: $C3 \times percent\_rank_{it}$

First actions are sorted based on the current SoC of the vehicle, the higher the SoC of vehicle the lower the rank, because we always want the highest charge vehicle to stop charging. Among actions concerning each vehicle (e.g. stopping charge now, stopping charge in 7.5 minutes, stopping charge in 15 minutes, and stopping charge in 22.5 minutes), actions are rank based on their proximity to the average stopping charge SoC in algorithm A, so if they are within 30 percent range around the average they get the lowest rank, and rank goes higher as they move further from that range.

Objective:

Minimize: $\sum_{iJ} x_{iJ}\left(c_{iJ}+c\_relocate\_soc_i\right)+\sum_{in} x_{in}\left(c_{in}+c\_charge\_soc_i\right)+\sum_{iJt} x_{iJt}\left(c_{iJ}+c\_stopcharge\_soc_{it}\right)$
$+c\_slow\_charge\_flow\times\sum_{IN} s\_slow\_charge\_flow_{IN}+c\_fast\_charge\_flow\times\sum_{IN} s\_fast\_charge\_flow_{IN}$
$+c\_in\_slow\_charge\times\sum_t s\_in\_slow\_charge_t+c\_in\_fast\_charge\times\sum_t s\_in\_fast\_charge_t$
$+c'\_in\_charge\times\sum_t(s'\_in\_slow\_charge_t+s'\_in\_fast\_charge_t)+c\_relocation\_flow\times\sum_{IJ} s\_relocation\_flow_{IJ}$
(3.71)

Subject to:

Note: $L_i = I$ is used to indicate that the vehicle i last drop-off location is in zone I, and $L_i = n$ is used to indicate that the vehicle i is in charger n.
The following equations will check that flows planned by algorithm B get satisfied:

For relocation flows, having slack will put a bound on maximum combined cost of charge spent on a relocation trip and penalty for SoC of vehicle.

$$\sum_{L_i=I} x_{iJ}+s\_relocation\_flow_{IJ}=B\_relocation\_flow_{IJ} \tag{3.72}$$

For going to charging flows, having slack will put a bound on maximum combined cost of charge spent on a going to charge trip and penalty for SoC of vehicle.

$\forall N \in$ set of slow charging station zone:

$$\sum_{L_i=I,n\ in\ N} x_{in} + s\_slow\_charge\_flow_{IN} = B\_charge\_flow_{IN} \tag{3.73}$$

$\forall N \in$ set of fast charging station zone:

$$\sum_{L_i=I,n\ in\ N} x_{in} + s\_fast\_charge\_flow_{IN} = B\_charge\_flow_{IN} \tag{3.74}$$

For flows from chargers, distance from chargers to zone centers is fixed, hence algorithm B already has had accurate estimation of cost for these trips, therefore, no slack is required.

$\forall N \in$ set of slow charging station zone:

$$\sum_{L_i=N,t} x_{iJt} \geq B\_from\_charger\_flow_{NJ} \tag{3.75}$$

$\forall N \in$ set of fast charging station zone:

$$\sum_{L_i=N,t} x_{iJt} \geq B\_from\_charger\_flow_{NJ} \tag{3.76}$$

The following will check that chargers are utilized as planned throughout the next step ($t = 1:4$). The time-step from algorithm B has been been divided into to 4 time-steps in algorithm C, to differentiate for example cases where a charger is used 5 minutes in a span of 30 minutes, versus 25 minutes in a span of 30 minutes. Without this divide, algorithm C can postpone sending vehicles to charge with no penalty, as long as the delay is each time less than the time-step of algorithm B. While in practice this may not happen often, it would imply that there is no longer a guarantee that plan by algorithm B is put to action. The surplus in charge is put in place to make a balance between two policies, first that number in charge should follow decision by B, and second that vehicles should reach a range of SoC before stopping to charge:

In the following equation: $n, m \in$ Set of slow charging station:

$$\begin{aligned}\sum_{k=0}^{t}\Big(\sum_{L_i=I,n} x_{in} \times a_{ink} - \sum_{L_i=m,J} x_{iJk}\Big) = B\_in\_slow\_charge - s\_in\_slow\_charge_t + s'\_in\_slow\_charge_t \\ - initial\_in\_slow\_charge - p\_in\_slow\_charge_t\,, \forall t \in \{1,2,\dots 4\}\end{aligned} \tag{3.77}$$

In the following equation: $n, m \in$ Set of fast charging station:

$$\begin{aligned}\sum_{k=0}^{t}\Big(\sum_{L_i=I,n} x_{in} \times a_{ink} - \sum_{L_i=m,J} x_{iJk}\Big) = B\_in\_fast\_charge - s\_in\_fast\_charge_t + s'\_in\_fast\_charge_t \\ - initial\_in\_fast\_charge - p\_in\_fast\_charge_t\,, \forall t \in \{1,2,\dots 4\}\end{aligned} \tag{3.78}$$

The following will check that capacity of chargers is not violated, and thus avoid queuing:

$$\sum_{k=0}^{t}\Big(\sum_{i} x_{in} \times a_{ink} - \sum_{L_i=n,J} x_{iJk}\Big) \leq capacity_n - initial\_in\_charge_n - p\_in\_charge_{nt}\,, \forall t \in \{1,2,\dots 4\} \tag{3.79}$$

The followings will check that each vehicle is assigned to one destination at most:

For vehicles not in charge:

$$\sum_{J} x_{iJ} + \sum_{n} x_{in} \leq 1\,, \forall i \tag{3.80}$$

For vehicles in charge:

$$\sum_{Jt} x_{iJt} \leq 1\,, \forall i \tag{3.81}$$

3.4. Choosing vehicle for a request (Algorithm R)

3.4.1. General scheme of algorithm R

The first and foremost job of algorithm *R* is to avoid accepting the requests that will make it impossible for the vehicle to safely get to charging station after servicing its passengers. Therefore, when *Assignment extension* receives k potential vehicles for a request, it has to check for each vehicle, if the current SoC of the vehicle is higher than the combined consumption of the new tour (the tour including the new request), and the trip from last point of the tour to its nearest charging station. As it is computationally expensive to find the closest charger every time a request is made, we pre-calculate the areas in the city, from which vehicles can get to a charger within 5, 10, and 15 kilometers. Then when the request is made we only need to check if the point of interest (last drop-off of the options) falls within the area of 5, 10, or 15 km (maximum) distance to charging station.

Before going to the algorithm that chooses the best vehicle among k vehicles returned by the *MaaS Dispatcher*, a review on dependencies of assignment and charging problem is presented. Consider two successive intervals (each of 30 min) the demand in first time-step requires n vehicles, and the second one $2n$ vehicles. Now imagine if at the beginning of the first time-step, there are n vehicles that can serve in one time-step and n vehicles that can serve in two time-steps. Then it is beneficial to give incentives to the assignment algorithm to use the vehicles with higher SoC (that can serve in both steps) in the first interval, rather than the vehicles with lower SoC. Because if the operator uses the lower charged ones in the first interval, they would be out of charge and it would not have enough vehicles for the second interval. Needless to say, limiting the set of vehicles in the first interval increases the immediate routing cost, but it could compensate by not rejecting customers in the second interval.

The core idea is that using the high SoC more frequently will result in a homogenized charge among the vehicles (e.g. having two vehicles with SoC of 30% rather than one with 60% and one 0%). As mentioned previously this can allow more requests to be satisfied, or provide more flexibility in time that vehicles must go to charge and that will potentially reduce the cost of electricity, under variable prices. It should be noticed that policy of using high SoC vehicles does not have the same efficiency in all situations. For example, if operator has much more charged vehicles than needed for the next time-steps, then we can choose high or low SoC vehicle for the requests, freely. Therefore, an indicator is developed (3.4.2) to suggest when the policy of choosing high SoC vehicles more frequently, is crucial for profitability of the operation.

The other dependency between assiment and charging accrues in cases where the vehicle has to charge, right after dropping off passenger in the request that is to be assigned to it. Then it is preferred if drop-off location was closer to a charging station. Therefore, the insertion cost of the request to current tour of the vehicle can be combined with go to charger cost to achieve a lower overall routing cost. This completes all SoC related considerations that algorithm needs to choose between vehicles for a request.

Algorithm 2 describes how the k options returned from the *MaaS Dispatcher* are evaluated and ranked based on their SoC. The target is to assign a generalized cost to each vehicle that combines the insertion cost, going to charger cost, and a cost associated with SoC of the vehicle, and then choose the best option.

3.4.2. Metric for choosing vehicles with high SoC

The metric should identify when it is critical to use the high SoC vehicles more frequently than the low SoC vehicles. Therefore, we expect a high metric value if three conditions are met at the same time. First condition is that remaining charge of fleet is not significantly higher than the energy consumption that we expect over the rest of the day. The second is that there is a nonuniform SoC distribution across vehicles, it is trivial that if there is no variation in the SoC of fleet, SoC will not effect the assignment decision. The third condition is that expectation of demand in the near future is high. It is indeed in peaks that we need more vehicles at the same time, and the difference between two situations, one with two vehicles at SoC of 30% and the other with one vehicle at SoC of 60% and one vehicle with 0% becomes critical. Equation to 3.82 to 3.84 presents the formulation with which we can derive such an indicator. The metric in 3.83 can be interpreted as the ratio of vehicles with SoC above x that should serve a trip to all vehicles with SoC above x.

Figure 3.5 shows this metric calculated for one day. It is clear that the metric becomes smaller as x goes higher, because the metric is based on the very least vehicles that have to be used from each group. Imagine if we have only one car with SoC above 90, we can simply not use that vehicle, however, at the same time, we might state that we need 70% of all vehicles above SoC of 50% to be active right now. In addition, if all vehicles have the same charge (e.g. at the beginning where they are fully charged) the metric would be high, while there is no preference between the vehicles. For that reason we have to normalize the metric via Equation

Algorithm 2 Choosing between vehicles

$driving\_range$ is driving range of the vehicles
$metric(SoC)$ is the function that maps SoC to the metric for using high SoC vehicles
w_{metric} is the weight to calibrate effect of the metric
Get k vehicles from the *MaaS Dispatcher* with the insertion order of pick up and drop-off within vehicles current tour
For each option get last drop-off coordinate before and after inserting d_k, and d'_k
For each option get existing tour T_k and SoC L_k
For each option calculate consumption cost of the tour before and after inserting c_k, and c'_k
for km in [5, 10, 15, 20] **do**
    if distance from d_k to nearest charge< km **then**
        $c\_tocharge_k = 100 \times \frac{km}{driving\_range}$
        break
for km in [5, 10, 15, 20] **do**
    if distance from d'_k to nearest charge< km **then**
        $c'\_tocharge_k = 100 \times \frac{km}{driving\_range}$
        break
if $L_k < c'_k + c'\_tocharge_k + 5$ **then**
    Eliminate option k
if $L_k - c'_k < 15$ **then**
    must charge after last drop-off $g_{k'} = 1$
Calculate generalized cost as $C_k = c'_k - c_k + g_{k'}(c'\_tocharge_k - c\_tocharge_k) + w_{metric} \times metric(L_k)$
Choose k with $minC_k$

3.84, to see the relative emphasis on using each SoC group x.

The idea behind normalization is that if the ratio of vehicles above x that should serve a trip is the same as ratio of the vehicles above x to all potential vehicles, then normalized metric should be one, suggesting that we do not need to promote the choice of vehicles with SoC above x. We only need to promote their frequent use if for example 20% of potential vehicles are above x, but 40% of them should be used. Therefore, metric is only put into effect if it is higher than one. We also see that at the start of the day, unlike Figure 3.5, all groups have normalized metric of one. The highest value accrues just before the peak where it is expected, and after that there are no vehicles with SoC above 80% for the rest of the day.

$$p_{xt'} = \frac{T_x - (T_0 + T_c - T_d)}{n \times d} \tag{3.82}$$

$$p_x = max_{t'}(p_{xt'}) \tag{3.83}$$

where:

$p_{xt'}$= Unnormalized metric for vehicles with SoC above x considering the horizon from t to t'
p_x= Unnormalized metric for vehicles with SoC above x
T_x= Trips all current vehicles with SoC above x can serve until time t'
T_0= Trips all current vehicles can serve until t'
T_c= Trips all vehicles that come out of charge can serve until t'
T_d= Trips the operator wants to serve until t'
n= Number of vehicles with SoC above x
d= $t' - t$

$$P_x = max\left(\frac{\frac{p_x}{p_0}}{\frac{n_x}{n_0}}, 1\right) \tag{3.84}$$

where:

P_x= Normalized metric for vehicles with SoC above x

p_x= Ratio of vehicles with SoC above x that should serve a trip to all vehicles with SoC above x (Unnormalized metric for vehicles with SoC above x)

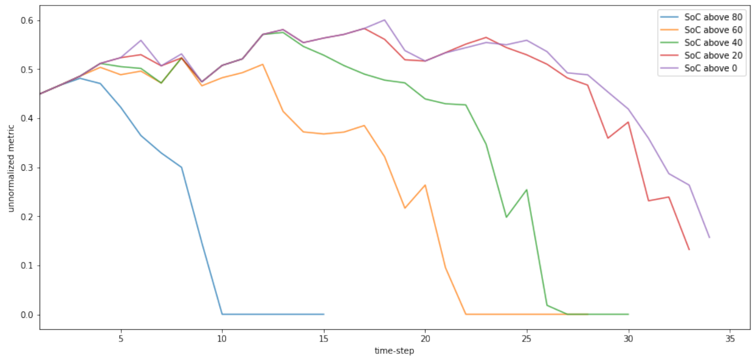


Figure 3.5: Example of unnormalized metric calculated for all time-steps

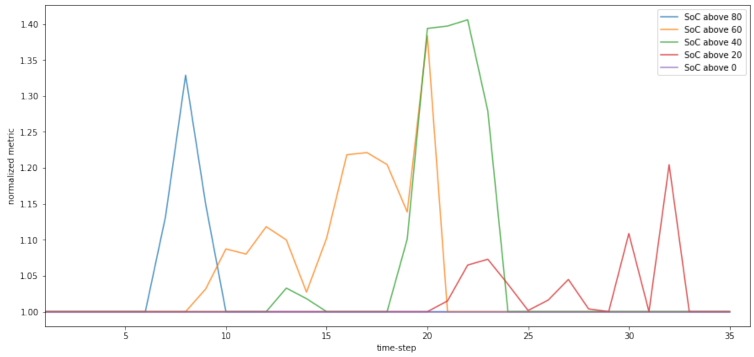


Figure 3.6: Example of normalized metric calculated for all time-steps

p_0= Ratio of all vehicles that should serve a trip to all vehicles that can serve a trip (Unnormalized metric for all vehicles)

n_x= Number of vehicles with SoC above x

n_0= Number of all vehicles that can serve a trip

3.5. Consumption function

The consumption function used in this study is only a function of the distance traveled. The reason is that routing service was outsourced to PTV xroute, and currently does not allow any direct access to the road network such as saving extra attributes on its nodes or links, to pre-calculate part of the consumption function. The only way to get information from the network is to query travel time for given sequence of locations. That is why adding extra variables to the consumption function would only increase computation time, and not improve algorithm in any meaningful way. With that said, we should be aware of the level of detail necessary in calculating consumption in each step of the algorithm presented. Fast ways to calculate consumption function remain a starting point for future work. Figure 3.7 will explain the variables that affect consumption, and whether we need to include them in the in each step. The darkness of the color in the figure is increasing with relation to accuracy required in calculating the consumption.

			Sources of variation in the consumption function				
Purpose	Required detail	Application in charging algorithm	**Link**	**Route**	**Time**	**Vehicle**	**Random**
			Grade, class, free speed	**Distance, travel time, elevation gain**	**Service level:** Speed, acceleration **Weather:** temperature, rain	**Battery age / driver aggressiveness**	**Everything unmeasurable, or unpredictable**
Prevent the battery from running out.	**Upper bound** is important.	**(R)** After each request consumption for the trip and to nearest charger after drop off, should be checked. **(C)** For sending vehicles to charger.	Grade, class, free speed	Not required if calculated link based. Could also be a pessimistic function of Distance, travel time, elevation gain	Time dependent safety factor	Factor for the vehicle	Safety factor
Estimating cost of empty routing trips	**Relative between vehicles.** Approximation is just as good.	**(C)** For a set of vehicles to a set of zone centers and chargers. This combinations come from algorithm B.	–	Distance, travel time, elevation gain.	Adapt zone-zone cost. It will not matter for individual vehicles since their travel time will change similarly, given origin and destination zone.	–	–
Checking SoC of vehicles by their last drop off	**Approximation** is just as good.	**(C)** For all vehicles that have a planned tour	–	Distance, travel time, elevation gain.	Zonal service level, Weather	Factor for the vehicle	–
Daily planning for charging trips	**Limited details.** Since demand stochasticity will still dominate the possible variation in the consumption.	**(A)** For estimation of average consumption per time step.	–	–	Network service level, Weather	Average state of fleet	–

Figure 3.7: Required accuracy in calculating consumption for each step of the algorithm

4
Case study: design and results

4.1. Case study design

4.1.1. Parameters of the problem

Five scenarios are designed in the case study for Barcelona, to assess the effectiveness of plans by the developed method. After the description of each scenario, the chapter will define the parameters mutual in all the scenarios, such as fleet characteristics, and the procedure of creating the requests and the demand estimates. Finally, exact steps for obtaining all required data for executing all of the scenarios are presented in Algorithm 3. Figure 4.5 summarizes the steps in Algorithm 3. In particular, it will show which request sets are shared between the five scenarios, and how the demand estimation, which operator would normally derive from historical operation data, is emulated for the case study.

- First is the base scenario, in which we take a simplistic approach to charging, also referred to as lazy charging for the remainder of the report. In lazy charging algorithm, vehicles only move toward the charger if their SoC goes below 20% and will then recharge in the closest charging station up to SoC of 90%. The vehicles will not go to the second closest charger if the first choice is full.

- Second is the smart charging scenario that includes using the *Charging and relocation extension* and the *Assignment extension*. This scenario will be the main focus of the case study.

- Third is the scenario where the assignment is independent of SoC. Meaning that we use the *Charging and relocation extension* but there is no longer a generalized cost calculated in the *Assignment extension*. The *Assignment extension* will only make sure that vehicles do not accept requests that they do not have enough charge for. The purpose of this scenario is to see the contribution of algorithm $\underline{R}$ and the metric developed for choosing high SoC vehicles more frequently.

- In the fourth scenario we want to see the performance of the method when the operator has overestimated the energy consumption throughout the day, and if the *Charging and relocation extension* can adjust the plan according to real-time information that it acquires. In regard to the method, it is the same as the smart charging scenario. The demand prediction also remains the same as the smart charging scenario, but the actual number of requests that realize through the day would be smaller by design.

- In the fifth scenario, we want to see the performance of the method when the operator has underestimated the energy consumption throughout the day. In regard to the method, it is the same as the smart charging scenario. The demand prediction also remains the same as the smart charging scenario, but the actual number of requests that realize through the day would be higher by design.

The parameters used to configure the scenarios are first explained and then summarized in Table 4.1.

Operation area: City of the case study in Barcelona. For which an OD matrix exist with 300 zones, that is aggregated over hours of the day. Figures 4.1, and 4.2 respectively show the spatial and temporal distribution of all trips in the city. Zones from the OD data have been clustered (Figure 4.3) and also bounded to Barcelona coast side (Figure 4.4), to be manageable with a fleet of 150 vehicles. Zones have to be aggregated, depending on the original resolution of the OD matrix, the size of the operation area, and fleet size. The zoning system is used in algorithm $\underline{B}$, which is designed to manage the aggregated flows of vehicles between the zones. Having

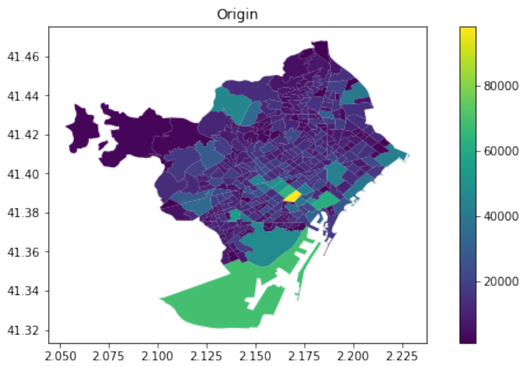


Figure 4.1: Number of trips originated from each zone over the whole day

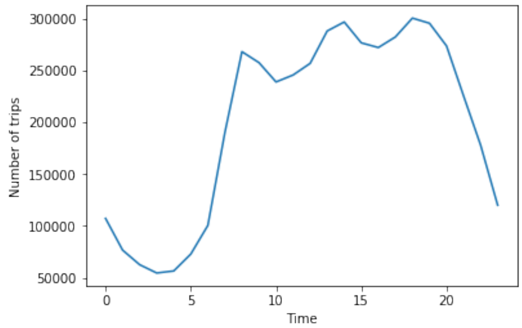


Figure 4.2: Temporal distribution of trips

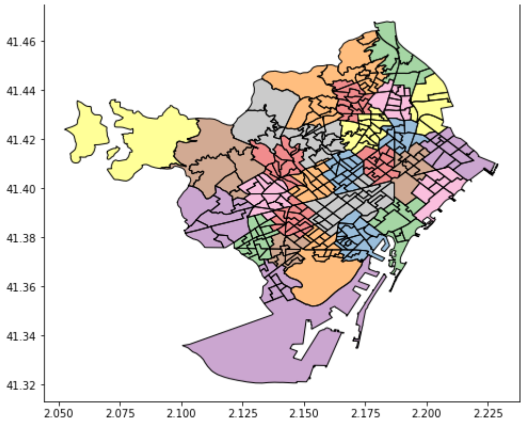


Figure 4.3: Clustering of the zones

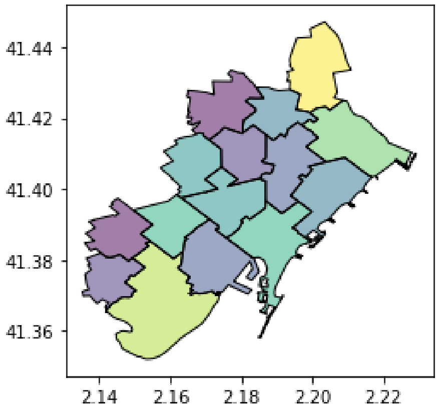


Figure 4.4: operation area for the case study

300 zones and 150 vehicles, is not compatible with algorithm B, as on average we would have half of a vehicle in each zone. The reader can revisit section 3.3.1 (last paragraph) for further implications of clustering on the algorithm. The method used to cluster the zones is Kmeans clustering [7], by using the latitude and longitude of the original zone centers as attributes of the cluster. The only tuning parameter of Kmeans clustering is the number of clusters, the 300 zones were grouped to 30 in the case study of Barcelona, from which 14 remained after reducing the size of the operation area. Figure 4.4 shows the zones used in algorithm B, while original zones are used to generate the trip requests for the simulations.

Fleet: The car used is Mercedes Benz evito, with a capacity of 6. The battery capacity is 40 kwh, which allows a fixed driving range of 120 km for this study, since the only variable in consumption function is distance. Charging the battery from 0% to 80%, with slow charging takes 6 hours, with the remaining 20% taking another 3 hours. Fast charging to 80% takes 1 hour, with the remaining 20% taking another 30 minutes.

Requests: For the remainder of the report, the term ground truth is used for the information derived directly from the OD matrix. For all scenarios we use the same expected total number of requests, that is set to n. Meaning in all scenarios we expect to see n requests. However, as it is explained next, no scenario will actually have n requests, due to the random process though which requests are generated. The total number of trips in the OD matrix, during the operation time considered in the case study, is N. We need to derive two sets of data from the OD matrix, for the case study. One for generating the trips, and one for generating perdition of pickups and drop-offs:

1. For the actual trips used in the simulation: First we need to obtain the temporal distribution of requests (number of trips per time-step). The number of trips per hour is derived from the ground truth by scaling the numbers from the temporal OD by $\frac{n}{N}$, and applying a white noise factor to each step. Besides, we can also introduce a bias in the morning or the afternoon, to mimic under and overestimations of demand (This feature will be used in the last 2 scenarios dealing with uncertainty of prediction inputs). During the next steps, the total number of trips per time-step obtained is held constant, this way an id

can already be assigned to each trip. Notice that at this point number of all requests can be different from n.

Next, a random timestamp is assigned to each trip. The origin and destination zones for each trip are drawn from a weighted sample of all Origin-Destination pairs (without replacement). Now we have the total number of trips with defined origin and destination zones, and yet need to allocate actual locations to each. For that purpose, two random numbers from 1 to 50 are assigned to each trip. They correspond to id of points in origin and destination zones, that had previously been generated for all zones. This means within each zone, there are 50 possible locations where a request can appear. Notice that, the expectation of number of pickups in each zone will be proportional to the one of ground truth, however this does not hold for each set of requests generated; for example, number of trips from zone 1 to zones 2 and 3 could be the same in the OD matrix, but not in a generated set of requests. Finally, the PTV, xroute service will automatically project this randomly generated coordinates to the closest on-road coordinate.

2. For demand prediction, namely the number of pickups and drops-offs per zone per time-step: Number of pickups and drop-offs expected is calculated by multiplying the ground truth by $\frac{n}{N}$. This data is subject to further modification explained in Algorithm 3 before being fed to algorithm B. In short, two auxiliary simulations are done without charging, to get the acceptance rate, the rate of shared trips, and the total distance traveled by the fleet. Then data that has only information on the requested trips is modified to account for the average number of requests that are shared or rejected, daily.

Table 4.1: Parameters of the problem in the case-study (applied to all scenarios)

Requests	**Unit**	**Base value**
Operation start time	time of the day	6
Operation end time	time of the day	22
Expected number of requests	count	8500
Average direct travel time	minutes	8
Max waiting time	minutes	10
Prebooking time	minutes	0
Max detour	dimensionless	1.6
Chargers		
Number of fast charging plugs	count	3
Number of fast charging stations	count	2
Number of slow charging plugs	count	20
Number of slow charging stations	count	4
Fleet		
Size	count	150
Capacity	count	6
Driving range	kilometer	120
0-80 slow charging	hours	6
80-100 slow charging	hours	3
0-80 fast charging	hours	1
80-100 fast charging	hours	0.5

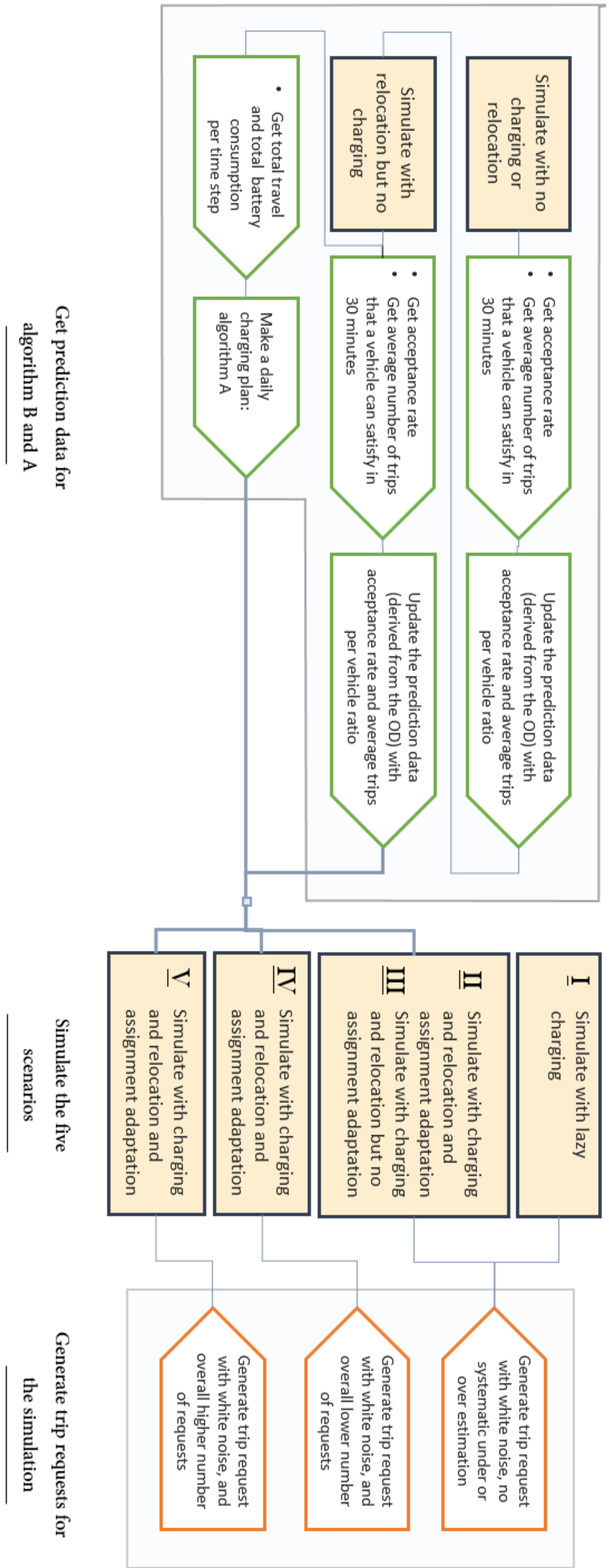


Figure 4.5: Summary of simulation steps

Algorithm 3 Simulation steps

Fix number of zone num_zones, number of requests $num\_requests$, number of cars num_cars, and location of charger stations. Set white noise factor w to 0, set morning and evening bias factors $b\_morning$, $b\_evening$ to 0
Generate a set of requests with origin-destination location and time-stamp
Simulate with no relocation and no charging trips (assuming unlimited battery)
Calculate the average number of pickups per time-step for vehicles that have at least one trip in the time-step, λ_t
Get number of estimated pickup and drop-off for each zone per time-step PO_{zt} and DO_{zt}, from the OD matrix.
Divide PO_{zt} and DO_{zt} by λ_t. (This is to account for the shared trips and short trips that allow a vehicle to have 2 or more pickups per time-step.)
Calculate average consumption between all zone pairs, including the charger zones (this is used in algorithm B).
Calculate consumption from all charging stations to all zone centers. This is used to prepare the costs of relocations from chargers (used for algorithm C).
Calculate areas (polygons) around each charger station, from points inside which the station is closer than 5, 10, or 15 km. Combine polygons of each distance group, to get areas with known maximum-distance-to-charger. This is used in algorithm R, to put an upper bound on distance of drop-off locations to their nearest charger.
Generate a set of requests with origin-destination location and time-stamp
Simulate with relocation, using PO_{zt} and DO_{zt}
Update λ_t
Calculate for each time-step, the ratio of $\sum_z PO_{zt}$ to the number of successful pickups from the simulation, γ_t
Divide PO_{zt} and DO_{zt} by $\lambda_t \times \gamma_t$.
Calculate total travel time per time-step TTT_t in seconds
Calculate number of vehicles in operation $d_t = \frac{TTT_t}{60\times 30}$
Calculate total consumption per time-step C_t
Calculate consumption rate $C = mean_t(\frac{C_t}{d_t})$
Calculate number of charged cars required per time-step $q_t = f(d_t)$
Get daily plan from A, including:

- number of cars that should be in charge [slow, fast] in each time-step Y_t
- number that should go to charge [slow, fast] for each time-step X_t

if Smart charging scenario **then**
    Set w to larger than 0, set $b\_morning$, $b\_evening$ to 0
    Generate a set of requests with origin-destination location and time-stamp
    Use PO_{zt} and DO_{zt} and daily charging plan, Y_t, X_t, and P_t previously calculated
    Simulate with relocation and charging trips, and using SoC in finalizing assignment choice
if Scenario with overestimation (underestimation) **then**
    Set w to larger than 0, $b\_morning$, or $b\_evening$ to positive and negative alternately
    Generate a set of requests with origin-destination location and time-stamp
    Use PO_{zt} and DO_{zt} and daily charging plan, Y_t, X_t, and P_t previously calculated
    Simulate with relocation and charging trips, and using SoC in finalizing assignment choice
if Scenario assignment independent from SoC **then**
    Use the request generate for the smart charging scenario
    Use PO_{zt} and DO_{zt} and daily charging plan, Y_t, X_t, and P_t previously calculated
    Simulate with relocation and charging trips, but not using SoC in finalizing assignment choice
if Scenario with lazy charging **then**
    Use the request generate for the smart charging scenario
    Simulate with the lazy charging algorithm and no relocation

4.1.2. Parameters of the optimization

All the parameters used to tune the algorithms are listed in Table 4.2. The physical meaning of the parameter if applicable, and the hard bounds on the value of the parameter are mentioned. Changing most parameters, will result in more satisfied demand in one direction (with more visits to the charging stations), and will result in higher cost-efficiency per passenger in the reverse direction (with fewer visits to the charging stations).

charging and relocation extension used in the case study will not re-optimize daily plan by algorithm $\underline{A}$. According to the Chapter 3 it should be re-optimized every 4 hours, however, it has a long solving time. While in real life the solving time is less than the 4 hours reoptimization interval, in simulation, during running algorithm $\underline{A}$, 4 hours may well pass in simulation time. In that case we have to reoptimize again once we get the results for the previous optimization.

Table 4.2: Parameter in the algorithms

Code	Title	Unit	Base value	Interpretation of value, and considerations
	Zone size			
Z1	Number of zones	count	14	Number of zones should not be larger than fleet size divided by 5
	Steps and horizons			
T1	A time-step duration	min	30	As low as the resolution is input data allows
T2	A reoptimization	hr	4	Bounded from below by running time of algorithm A
T3	B time-step duration	min	30	As low as the resolution is input data allows, bounded from below by average time between two pickups
T4	B horizon	steps	5	Longer than average charge duration, bounded from above by computational complexity
T5	B reoptimization	sec	240	Depending on the emergence of new information, it can decrease by an increase in fleet size
T6	C reoptimization	sec	T5	Depending on the emergence of new information, it can decrease by an increase in fleet size
T7	C step for in charge constraint	sec	T3/4	More than 5 minutes
	Weights for A			
A1	An active vehicle	percent charge	150	Average profit from 30 minutes operation of a vehicle, divided by the average price of a percent charge
A2	A charged vehicle	percent charge	20	Less than A1, the exact value should be calibrated for best results
A3	Going to slow charging station	percent charge	10	Average charge spent on a trip going to a slow charger, plus opportunity loss from being on the way
A4	Going to fast charging station	percent charge	20	Average charge spent on a trip going to a fast charger, plus opportunity loss from being on the way
	Weights for B			
B1	Penalty for unsatisfied pickup	percent charge	7	Maximum charge spent on a relocation trip
B2	Penalty for slack in slow charge	percent charge	8	Should be higher than B2
B3	Penalty for slack in fast charge	percent charge	10	Should be higher than B2
B4	Penalty for slack going to slow charge	percent charge	10	maximum charge allowed for trip to charging and back plus B2
B5	Penalty for slack going to fast charge	percent charge	12	maximum charge allowed for trip to charging and back plus B2
B6	Penalty for surplus in charge	percent charge	3	Positive value larger than the charge spent on any trip from chargers to the nearest zone
	Weights for C			
C1	Penalty for 1 percent higher SoC for vehicles going to charge	dimensionless	0.15	If SoC of a vehicle is higher by 1 percent, it will be chosen if its consumption for the charging trip is up to C1 percent higher
C2	Weight for metric for SoC group of vehicle relocation	percent charge	-4	If metric for a vehicle is higher by 1 increment, it will be chosen if its consumption for the relocation trip is up to abs(C2) percent higher
C3	Penalty for percent rank of the vehicle-action in its category of possible actions	dimensionless	4	This will control that vehicles stop charging when they have reached the SoC level planned by daily charging planner
C4	Penalty for slack relocation flow	percent charge	B1-C2	It will control the maximum charge spent for relocating a vehicle, depending on its SoC
C5	Penalty for slack in slow charge for 5 minutes	percent charge	0.5	less than charged gained in 5 minutes with slow charger
C6	Penalty for slack slow charging flow	percent charge	$\frac{B2}{2} + 50 \times C1 - 2 \times C5$	It will control the maximum charge spent for sending vehicle to slow charge
C7	Penalty for slack in fast charge for 5 minutes	percent charge	4	less than charged gained in 5 minutes with fast charger
C8	Penalty for slack fast charging flow	percent charge	$\frac{B3}{2} + 50 \times C1 - 2 \times C7$	It will control the maximum charge spent for sending vehicle to fast charge
	Weights for R			
R1	Weight for metric for SoC group of vehicle	percent charge	-4	If metric for a vehicle is higher by 1 increment, it will be chosen if its consumption is up to abs(R1) percent higher
R2	Weight for insertion cost	dimensionless	1	Is set to one, R1 and R2 are set relative to this value
R3	Penalty for charge spent from last drop off to charging station	dimensionless	0.8	Marginally less than R2, because it happens later in the future

4.2. Performance of algorithms

Experiments to report performance were run on a computer with 8 Intel R processors running at 3600 MHz using 32 MB of RAM, running Windows version 10. The performance was tested for fleet sizes of 150, 1500, and 15000. The problems were constructed by expanding the problems from the simulation with 150 vehicles. The problems in B and C vary in complexity along the time-steps in the simulation, the problems represented here have been selected from peak times of charging. Table 4.3 and 4.4 suggest that algorithm A's, and B's solve time remains independent of fleet size, and algorithm C's will grow linearly. Time to solve algorithm B and C are interdependent, and rely on the size (and consequently number) of the zones. The larger the size of the zone (the smaller the number) algorithm B becomes less computationally costly at the expense of a longer preparation and solve time for algorithm C.

In addition to the *time construct problem*, and *time solve problem* of algorithm C, in Table 4.5, time to get the battery consumption (from *xroute*) for the vehicle-destination pairs considered in algorithm C should also be added to the computation cost. They could not be reported for larger fleet sizes since they could not be replicated without running the simulation with the same fleet size. However, it is expected that the number of the options that we need to calculate consumption for, would grow faster than linear with the number of vehicles in the zone, as the number of available vehicles are multiplied by the possible destinations.

Reader should recall that algorithm A was solved with Column Generation in which first we make many plans for individual vehicles and then choose how many vehicles should follow each of the plans. Solve time of algorithm A can be further reduced by saving the best plans of each day and using them for successive operation days, as initial columns in the master problem. Furthermore [4] offers two main techniques for accelerating Column Generation by bounding the cost in the sub-problem, and reducing the number of iterations required in the Column Generation. These should be implemented in the future work.

Table 4.3: Performance of algorithm A

fleet size	objective	time (seconds)
150	-3.13E+05	499.86
1500	-3.15E+06	453.17
15000	-3.13E+07	462.47

Table 4.4: Performance of algorithm B

fleet size	objective	gap	time to construct problem (seconds)	time to solve problem (seconds)
150	1560.33	0.083	0.043	1.596
150	1618.12	0.083	0.096	2.370
150	1235.3	0.089	0.041	1.017
150	1304.92	0.098	0.039	2.656
1500	13648.2	0.024	0.040	0.102
1500	14277.9	0.023	0.119	0.460
1500	11471	0.064	0.053	0.196
1500	11277	0.018	0.056	0.967
15000	133084	0.002	0.095	0.163
15000	139389	0.005	0.040	0.226
15000	112571	0.049	0.050	0.114
15000	110177	0.001	0.105	0.739

Table 4.5: Performance of algorithm C

fleet size	objective	gap	time to construct problem (seconds)	time to solve problem (seconds)
150	186.273	0.000	4.971	0.005
150	160	0.000	5.483	0.003
150	91.8565	0.000	6.450	0.008
150	110.079	0.000	6.422	0.003
1500	1863.1	0.000	6.319	0.046
1500	1600	0.000	6.291	0.025
1500	911.276	0.000	8.170	0.035
1500	1023.17	0.000	8.405	0.040

4.3. Case study results

First we discuss the main differences in the key performance indicators presented in the Table 4.6 among the scenarios (Section 4.3.1). Then, beginning with the next (Section 4.3.2) we go in more detail, scenario by scenario, and explain the mechanisms that generate the values presented in the table.

4.3.1. Comparison between the scenarios

According to Table 4.6, the smart charging scenario had 8% higher acceptance rate than the lazy charging, which means using smart charging resulted in 14% more revenue for the operator. The higher acceptance rate is a joint result of relocating and having fewer rejection owing to not having enough charge. The number of rejected requests due to not having charge decreased from 702 to 238 (464 fewer) with smart charging, and overall 612 more requests were satisfied. Reader should notice that had the operator relocated vehicles in the lazy charging scenario, the number of rejected requests in the lazy charging scenario would have increased even more.

Distance per request and ratio of travel time to direct travel time, are both measures of cost-effectiveness. As expected lazy charging is the best (lowest) in these measures, for which it compromises overall lower acceptance rate. The worst, also as expected, accrues when the operator overestimate the demand, as it prepares for something that never realizes. When the assignment is independent of the SoC, the algorithm always chooses the best vehicle solely by measure of the insertion cost, thus decreasing the cost per passenger. The results are aligned with this expectation.

Percent of relocation who got a trip after, is calculated by flagging the relocations as successful if the vehicles got a trip within 20 minutes from the time they were requested to relocate. As expected, it is lower if the operator overestimates the demand, however, that is also because with fewer requests, vehicles are less likely to get a trip even if they do not relocate.

Total relocation distance is inflated in the scenario results since the Simulator cannot currently handle deleting a relocation request after the vehicle gets a trip, while not yet arriving at the relocation destination. Normally, the relocation should be canceled if a vehicle gets a request. Relocation distance is about 5% of the total distance traveled. We expect that the overall time to drive to passengers pickup-location would decrease with the 5% that we spend on relocation, however, that information could not be extracted from the Simulation.

Number of charging trips increases substantially with the use of smart charging. That is firstly because smart charging gains more charge, for that, it has to use chargers earlier on in the day when vehicles battery are not yet empty. The second reason is that even when vehicles have the capacity to charge more, smart charging prefers to distribute the charge between the vehicles. Unexpectedly, the number of charging trips is even more in the scenario where we have overestimated the demand, but have less real charging needs. This is because of the policy of keeping vehicles in charge until they reach a certain SoC. In smart charging scenario and underestimation scenario, this policy results in keeping the vehicles in charge for longer, and thus fewer new vehicles can go to chargers, due to the limited charger capacity; however, in overestimation vehicles already begin with a higher SoC when they go to charge, and can soon be discharged from the charger, making way for new vehicles. So in overestimation scenario we have less charge gained while more trips are made to the chargers. This is an adverse effect and should be corrected in future work. It may be prevented by choosing a higher upper bound for SoC of vehicles that can go to charge, and by reoptimizing algorithm $\underline{A}$.

Average consumption to charger and back is around 7 km in all scenarios using smart charging algorithm, and is 2 km in the lazy charging scenario. The difference can be explained with three factors. First, average consumption to charger and back in all scenarios except lazy charging, is technically a charging and a relocation trip, as by the end, vehicle ends up where the demand is expected; on the other hand, in lazy charging vehicle is left near the charger after it reaches 90% SoC, waiting to get a passenger. The average distance of inbound trips to chargers, in the smart charging scenario, was 2.7 km. Second, smart charging sends more vehicles to charge, this alone increases the average cost, as the extra vehicles should be moved from areas further away from the chargers. The last reason is that in lazy charging most vehicles reach below 20% SoC at the same time, and the algorithm can choose from many vehicles to send to charge, which decreases the average cost, while in smart charging, algorithm tries to find the lowest charged car to send to charger. The total distance to charger and back is about 7 % of the total distance traveled.

Ratio charge spent to charge gained highlights the downside of the frequent short visits of the scenarios with smart charging compared to the lazy charging scenario. To use the full capacity of all chargers, vehicles have to travel longer. Lazy charging never uses all of charger even though there is queue in other chargers, because vehicles only go to the closest one. It is important to distinguish that the undesirable low ratio of charge gained for charge spent is not a property of the algorithms, but the objective weights, operator, shall he want, can opt for less charging, with risk of losing some customers.

Table 4.6: Results

	Unit	Lazy charging	Smart charging	Assignment independent of SoC	Over estimation	Under estimation
Total trips	count	7740	7740	7740	5642	9997
Accepted requests	count	4293	4905	4839	4092	5297
Total rejected requests	count	3347	2835	2901	1550	4700
Rejected for not having charge	count	702	238	261	0	1358
Percent accepted	percent	55.40	63.37	62.52	72.53	52.99
Total distance	km	16199	20437	19318	16547	21835
Total travel time	hour	848	1061	1009	862	1128
Distance per request	km	3.77	4.17	3.99	4.04	4.10
Ratio total travel time to sum of direct travel time	ratio	1.25	1.46	1.4	1.51	1.43
Num relocations	count	na	403	388	335	354
Avg relocation distance	km	na	2.7	2.5	2.4	2.6
Total relocation distance	km	na	1104	955	799	937
Percent of relocation who got a trip after	percent	na	78	77	65	80
Number trip to charger	count	32	213	210	282	195
Avg trip distance to charger and back	km	2.1	7.0	7.1	7.0	7.1
Total distance to charger trips and back	km	67	1431	1431	1942	1327
Total time in charge slow	hr	65	164	162	153	179
Total time in charge fast	hr	12	29	28	26	29
Electricity cost penalty	na	1388	4302	4143	3727	4815
Ratio charge gained in charging to charge spent on charging trip	ratio	35.1	3.9	3.8	2.46	4.55
Total charge gained in slow charging	in ratio to one full battery	9.4	24.4	24.0	21.6	27.0
Total charge gained in fast charging	in ratio to one full battery	10.0	21.8	21.0	18.1	23.0

4.3.2. Smart charging scenario

SoC evolution, and trip acceptance

Figure 4.7 shows how SoC of fleet evolves in the day. The 150 vehicles start with a full battery, and their SoC decrease homogeneously, before the first charging trips. SoC of vehicles going to charge is shown in orange and SoC of vehicles coming out of charge is shown in green. The vertical distance between the green and orange shows the average charging duration throughout the day. The algorithm is good at choosing low SoC vehicles to go to charge, which does in return increase the average cost of charging trips. SoC at which vehicles stop charging seems also to follow the rule of algorithm A. The green markers at a higher SoC in comparison to the majority of green markers, are vehicles kept in fast charging stations for longer, waiting for a vehicle to replace them, so that number of in charge vehicles compile with algorithm A.

In Figure 4.7, from hour 13 (equivalent to time-step 26 in Figure 4.6), vehicles start running out of charge, by the end of the day, 40 vehicles have SoC below 15%, resulting in the darker blue color, at the bottom right corner of the figure.

The Figure 4.6 shows that in the early hours fleet keeps up well with the demand, and as demand increases the number of satisfied trips reaches a saturation level. Starting from time-step 26 requests get rejected because of having low battery. However, the green line in this figure (and in figures 4.23, 4.29, and 4.33) is misleading. Meaning that even if the fleet had enough charge it could still not satisfy all the trips in the green line, because if the vehicle had accepted the first request, it would have not been in reach for the second potential request, however now, since first one is refused for not having charge, vehicle gets more offers. In all the mentioned figure,s the reader can pay attention to the increase in the gap between requested and satisfied trips. The escalation in the gap between blue and orange line is a better indicator of how many potential customers the operator is losing because of charge limitation. We can also observe a slight decline in trips satisfied around time-step 15, which coincides with the peak charging time of vehicles.

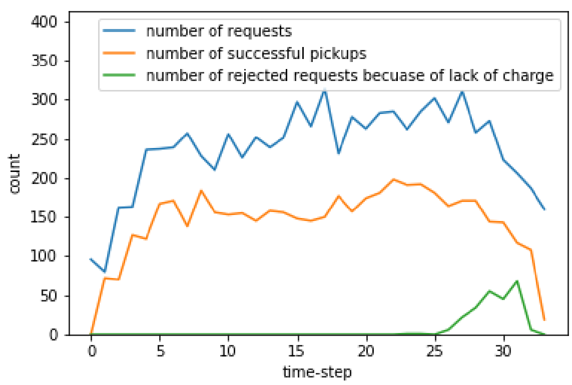


Figure 4.6: Trips

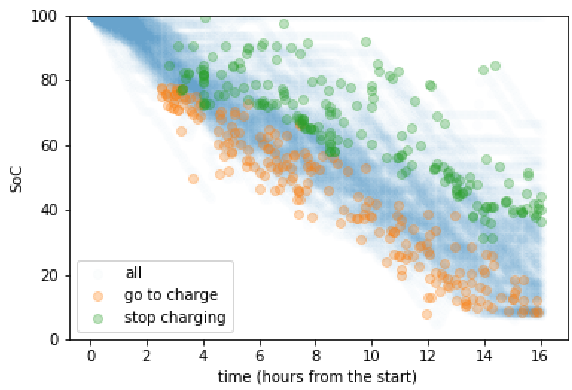


Figure 4.7: SoC of vehicles in time

Statistics per vehicle

Figures 4.8, 4.9, 4.10, and 4.11 show the distribution of demand among the vehicles. Most vehicles have 30 to 45 trips a day, few vehicles have less than ten trips, which should be a result of stochasticity in demand, and the relocation algorithm not being suited for that. Figure 4.8 shows that in 35% of total vehicle-hours there were no pickups. The maximum number of pickups in 30 minutes was 4. Figure 4.10 shows the maximum number of trips to charger was 4, which could be irritating for the drivers, meanwhile 12% of all vehicles had no charging trip at all. Figure 4.11 shows number of relocation trips, which ranges mostly between 1 to 4, translating to 1 relocation trip for every 13 passenger trips.

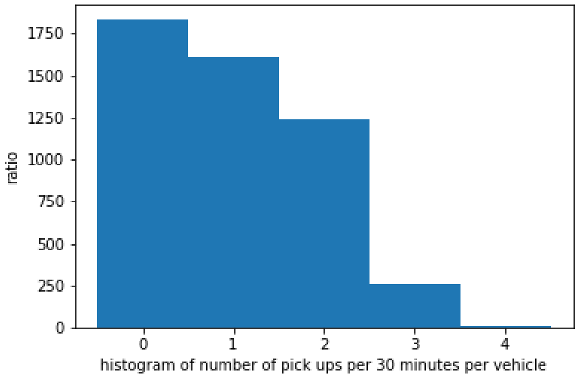


Figure 4.8: Histogram of number of pick ups per 30 minutes per vehicle

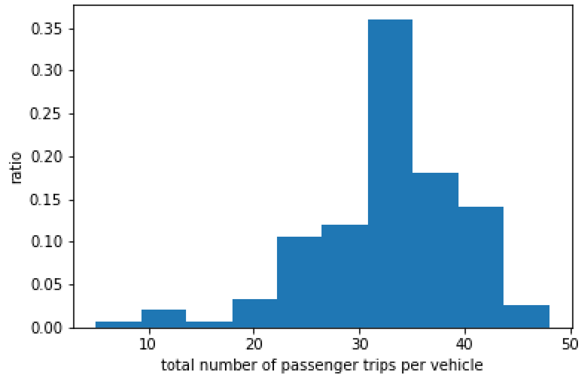


Figure 4.9: Histogram of total number of passenger trips per vehicle

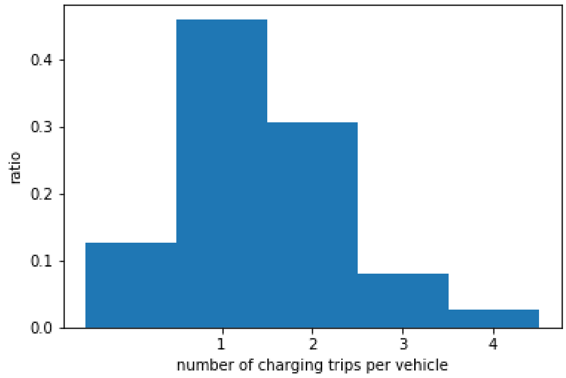


Figure 4.10: Histogram of number of charging trips per vehicle

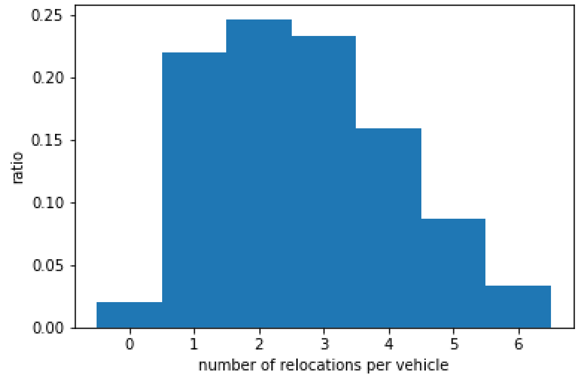


Figure 4.11: Histogram of total number of relocation trips per vehicle

Chargers and charging trips

The pink line in figures 4.12 to 4.15 show the daily plan by algorithm A. While fast chargers are used almost to capacity after time-step 13, there is more fluctuation in plan for slow chargers. This could be a response to peak electricity prices that accrue from time-step 8 to 13 and 17 to 23. It could also be attributed to the plan of algorithm A not being optimal. It seems as though algorithm A could push charging trips later in time, avoiding to send vehicles to charge that have high SoC such as 70%. Charging duration in some time-steps also seems too short to be in the optimal plan. These are speculations from the results, as the gap of the solution in algorithm A is in fact unknown.

The figures also compare the plan of algorithm A to the outcome of the simulation. Because of the high penalty of slack variables, the outcome is close to the plan by algorithm A. There is a delay in vehicles reaching charger, as both algorithms A and B assume that vehicles can immediately arrive at the charger. There is a lag coming out of charge because of the policy to penalize stopping charge outside the planned range. Figure 4.14 reveals a gap between the plan of algorithm A and the actual number of vehicles going to the slow chargers. As mentioned earlier, this is a consequence of the policy of keeping the vehicles in charge until they reach a specified SoC (also coming from algorithm A). The gap enlarges in time-steps where chargers are working close to capacity, meaning that new vehicles wanted to go to charge but the capacity of chargers did not allow, as previous vehicles were kept in charge due to the mentioned policy.

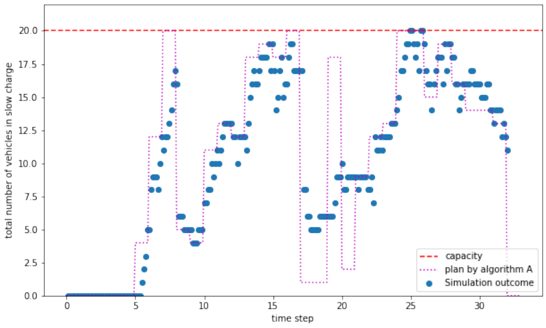


Figure 4.12: Number of vehicles in slow charge

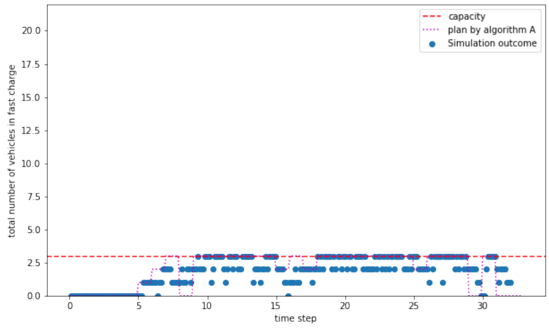


Figure 4.13: Number of vehicles in fast charge

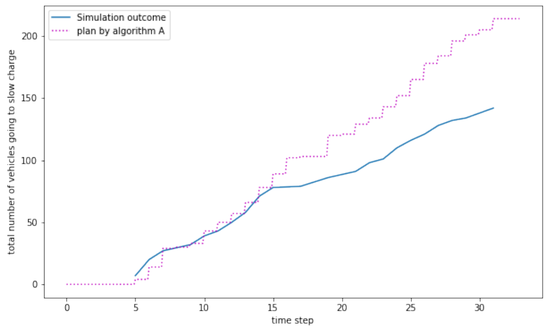


Figure 4.14: Number of vehicles going to slow charge

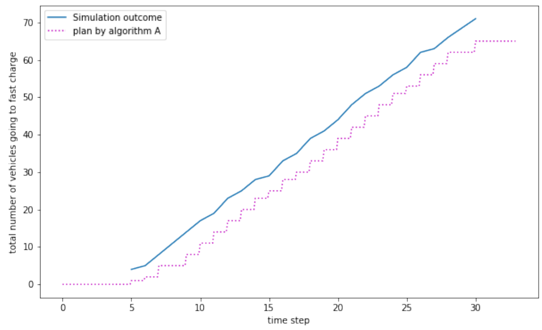


Figure 4.15: Number of vehicles going to fast charge

Figure 4.17 and Figure 4.18 show the range for the cost of charging trips. Most vehicles spend 1 to 3 percent charge going to the chargers. There are also a few less plausible values, up to 6%. That is because algorithm C does not have a bound directly on charge spent on charging. According to algorithm C if a vehicle has much lower SoC it may be longer on the way to go charger. We can add external rule to eliminate all charging trips above certain threshold, but then *Assignment extension* also has to be updated not to allow vehicles end up in areas where distance to closest charger is above the threshold, if their SoC is critical. Figure 4.16 shows SoC distribution of vehicles that started to charge.

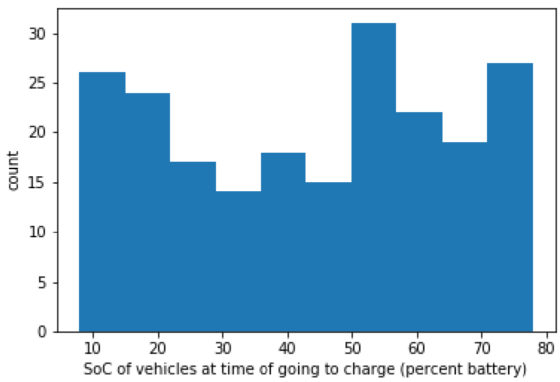


Figure 4.16: Histogram of SoC of vehicles at time of going to charge

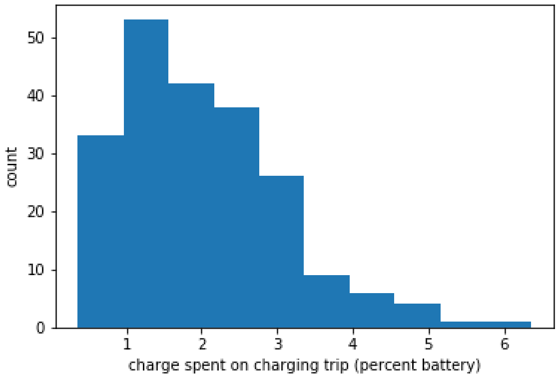


Figure 4.17: Histogram of charge spent on charging trips

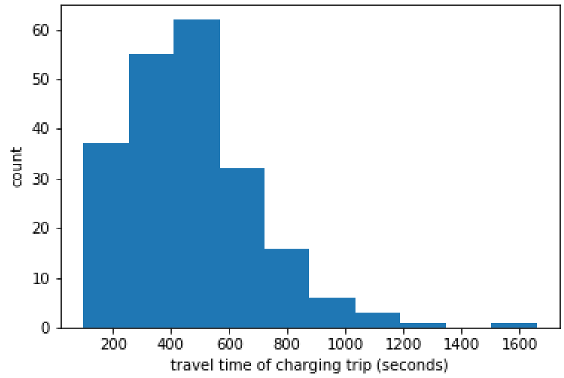


Figure 4.18: Histogram of travel time of charging trips

Results from the assignment extension (algorithm R)

In this section, we compare the original choice of *MaaS Dispatcher* to the vehicle choice of algorithm R. *MaaS Dispatcher* makes a decision only based on the insertion cost which is a combination of routing cost, waiting time, and detour factor. While algorithm R receives the ranking of vehicles by insertion cost calculated by *MaaS Dispatcher*, the actual scores remain confidential. Therefore, the insertion cost is recalculated just based on the routing cost, which is assumed to correlate with the *MaaS Dispatcher* insertion cost that we do not have. However, the rankings based on the two insertion costs are not always the same.

This is seen in Figure 4.20, had the ranking been the same no choice from *MaaS Dispatcher* would have had a higher insertion cost, than that of algorithm R. The consequence of using this insertion cost is an overall less routing cost, at the expense of increasing waiting time and detour for passengers, that is now hidden from the objective.

Algorithm R only deviates from the option with the lowest insertion cost under two conditions. One if another vehicle has a higher SoC, and two, if another vehicle has SoC close to 15% and the request being considered, brings its last drop-off point closer to charging station. Then logically, for vehicles with SoC well above 15%, *MaaS Dispatcher*'s choice should not have a higher SoC than of the algorithm R's. The markers in Figure 4.22, in the lower triangle with SoC well above 15% are the cases where the insertion cost calculated by the algorithm R and *MaaS Dispatcher* resulted in different rankings. We can see in Figure 4.21 that in the first time-step, 80% of choices were the same; we can conclude that the ranking is about 80% of times the same, as we expect the insertion cost to dominate the generalized cost of algorithm R in the first time-step, where all vehicles have almost the same SoC.

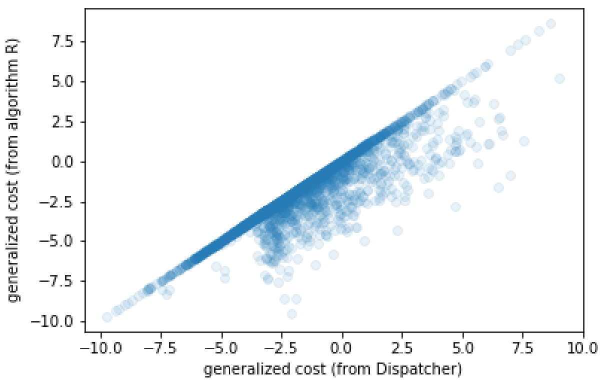


Figure 4.19: Generalized cost of vehicles chosen by algorithm R compared to vehicles chosen by the Dispatcher

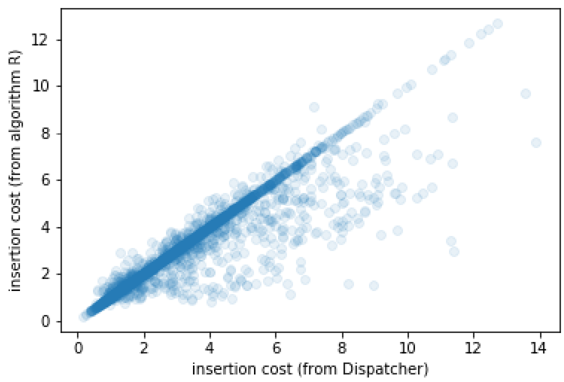


Figure 4.20: Insertion cost of vehicles chosen by algorithm R compared to vehicles chosen by the Dispatcher

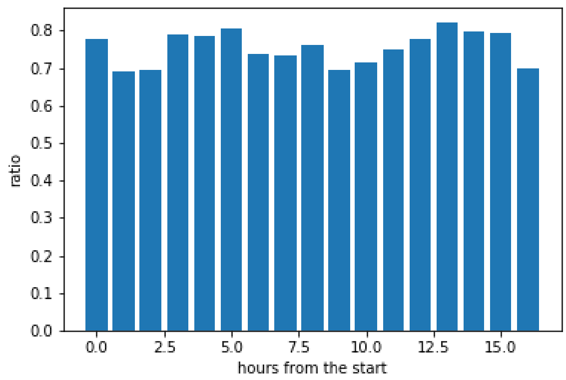


Figure 4.21: Ratio of trips who had the same choice of vehicle by Dispatcher and algorithm R

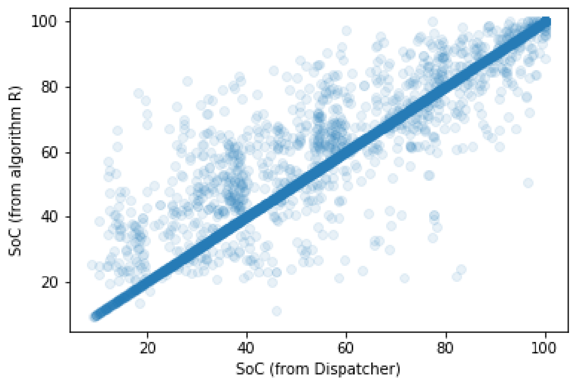


Figure 4.22: SoC of vehicles chosen by algorithm R compared to vehicles chosen by the Dispatcher

4.3.3. Lazy charging

Figure 4.23 shows that in lazy charging operator starts rejecting customers for not having charge from time-step 20, almost 2 hours earlier than in the smart charging, and finally a significant gap emerges between blue and orange line towards the end of the day. In the last time-steps in Figure 4.24 we see that many vehicles have battery above 50%, either because they charged to 90%, or they were not used from the start, while 75 vehicles accumulate in the bottom of the figure, with SoC less than 15%.

As Figure 4.25 shows, lazy charging start sending vehicles to charge the same time it begins to reject customers for not having charge. It makes no priority between the fast or slow chargers. It fails to use the chargers that were further away from the majority of low charged vehicles. That is why it will not use the slow chargers to their full capacity. This will always depend on the network of charging stations. Lazy charging was able to reach full capacity in a different simulation with a different distribution of chargers. Figure 4.27 and 4.28 show charge and time spent on charging trips, respectively. The closest charger in lazy charging was based on the air distance, and independent of the road network. The higher distances to chargers could be a result of not choosing the closest charger correctly, or vehicles truly ending up far from chargers while having SoC below 20%.

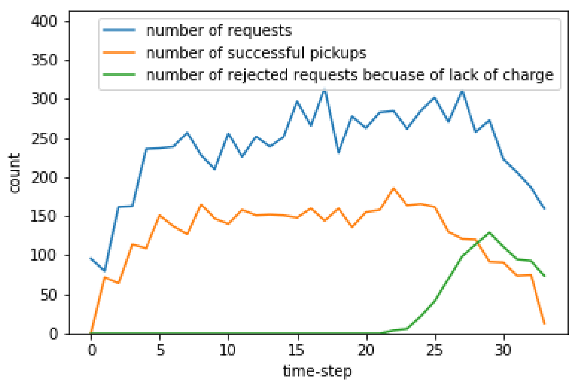


Figure 4.23: Trips (lazy charging)

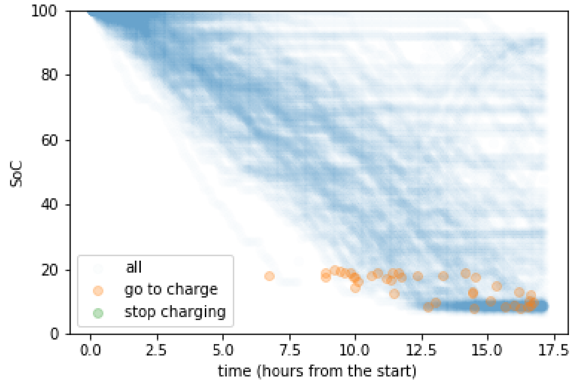


Figure 4.24: SoC of vehicles in time (lazy charging)

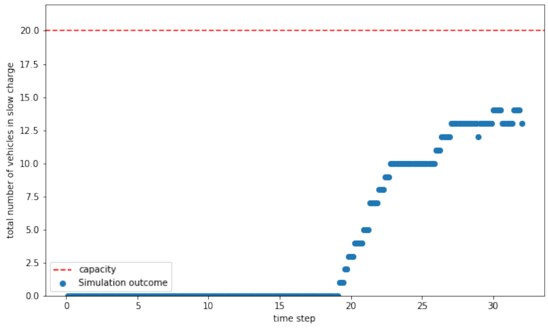


Figure 4.25: Number of vehicles in slow charge (lazy charging)

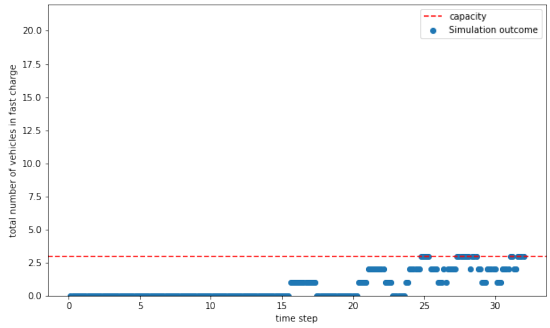


Figure 4.26: Number of vehicles in fast charge (lazy charging)

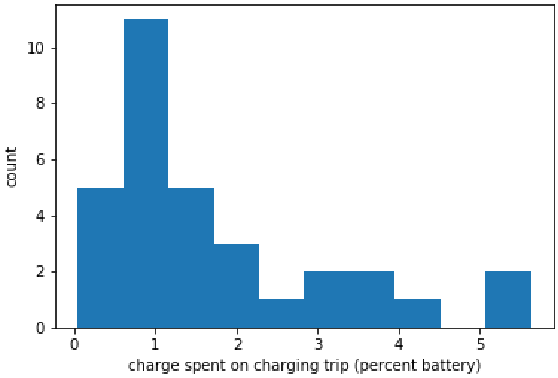


Figure 4.27: Histogram of charge spent on charging trips (lazy charging)

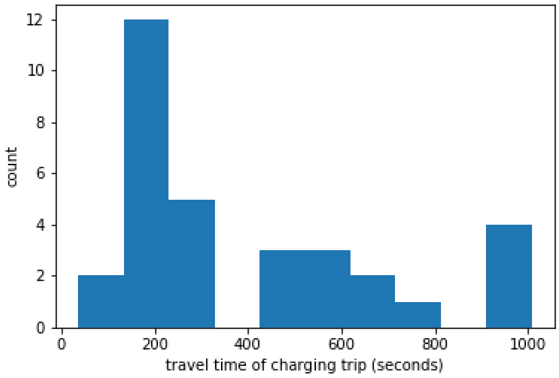


Figure 4.28: Histogram of travel time of charging trips (lazy charging)

4.3.4. Overestimation of demand for charging

In Figure 4.29 we can see the number of satisfied trips (orange line) surpasses the total requests (blue line). This is because the blue line is based on the time that passengers requested the trip, whereas the orange line is based on the pickup time of the passenger, which can be one time-step later. The gap between total requests and satisfied requests through the day is on average smaller than in the smart charging scenario, where overall requests were higher.

In Figure 4.30 we can almost observe two clusters (one for each charging peak) where SoC of going to charge (orange markers) and SoC of stopping charging (green markers) are close to each other, which means charging duration was short. This is the main setback of the algorithm for when the operator has overestimated the energy requirement of the day. The *Charging and relocation extension* should be able to modify the plan of algorithm $\underline{A}$ not only by reoptimizing it, but also to allow flexibility in each time-step to decrease the number of vehicles going to charge based on the average of SoC of the fleet. The number of vehicles in charge is more responsive to unpredicted changes in the SoC, as seen in Figure 4.31. In the overestimation scenario number of actual vehicles in charge always stays lower than the plan by the algorithm $\underline{A}$ (the pink line), unlike the smart charging scenario where we observed that the number of vehicles in charge can be higher than the plan by the algorithm $\underline{A}$.

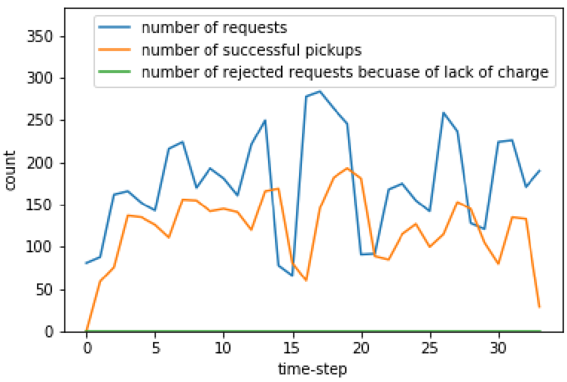


Figure 4.29: Trips (overestimation)

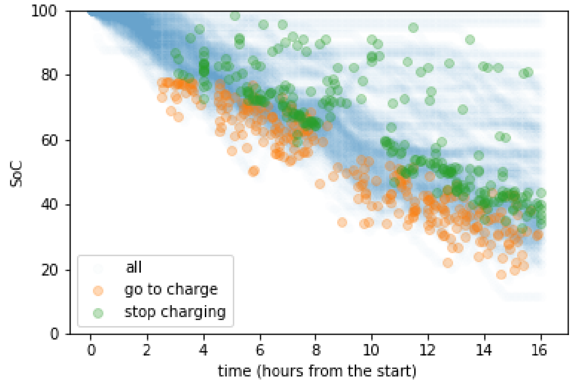


Figure 4.30: SoC of vehicles in time (overestimation)

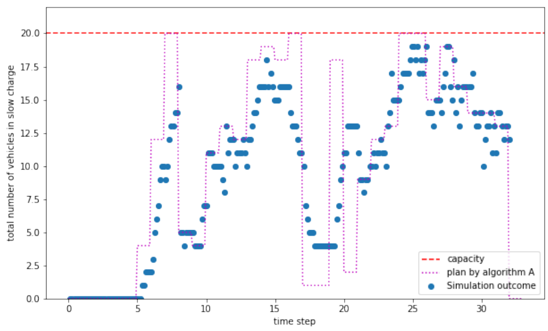


Figure 4.31: Number of vehicles in slow charge (overestimation)

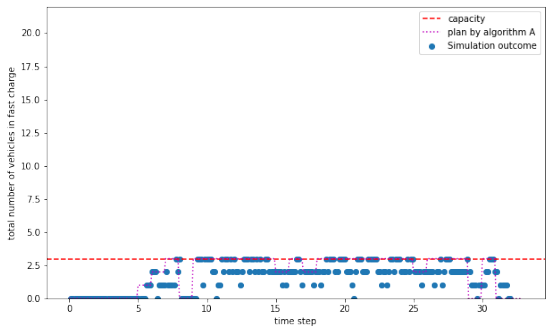


Figure 4.32: Number of vehicles in fast charge (overestimation)

4.3.5. Underestimation of demand for charging

In this scenario, the number of actual trips were higher, while the daily charging plan was the same. As expected vehicles run out of charge earlier in the day, and during hour 12 and 16, 80 vehicles have SoC below 15%. The gap between total and satisfied vehicles increases from time-step 25, and does not recover until the end of the day. Number of in charge vehicles are marginally higher than in the smart charging scenario, yet not enough to keep the same average SoC in the fleet as planned by algorithm A. Readers should notice that chargers are already working close to capacity, especially the fast chargers, and that there is bound to the number of vehicles operator can satisfy, regardless of the charging planner.

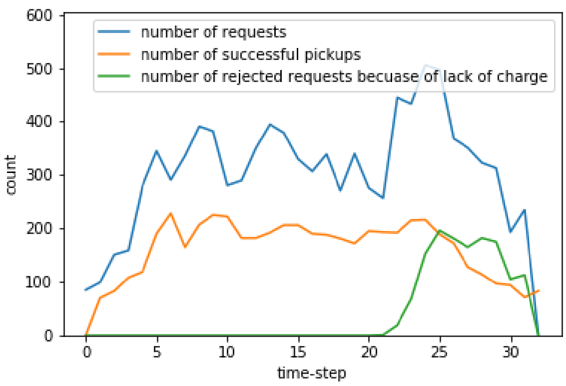


Figure 4.33: Trips (underestimation)

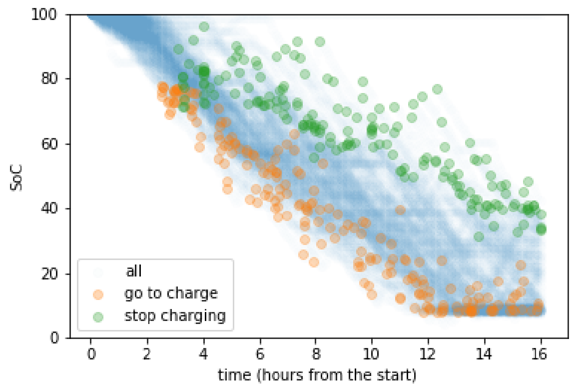


Figure 4.34: SoC of vehicles in time (underestimation)

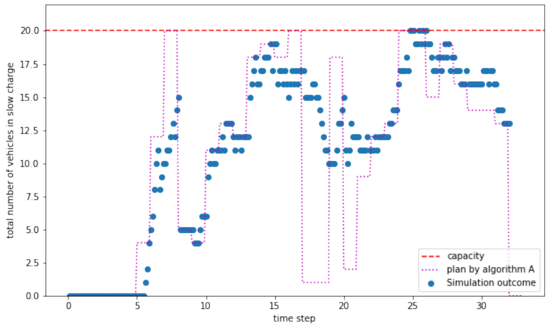


Figure 4.35: Number of vehicles in slow charge (underestimation)

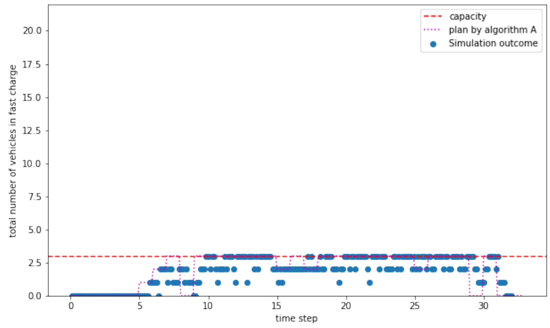


Figure 4.36: Number of vehicles in fast charge (underestimation)

4.3.6. Assignment with and without SoC as deciding factor

Figure 4.38 shows the SoC of vehicles evolve more heterogeneously than in the smart charging scenario. The maximum SoC is higher, and the minimum SoC is lower, thus operator start earlier to reject request for not having charge. However, as the maximum SoC begins to decrease after time-step 25, the total number rejected in the scenario is not significantly higher than in the smart charging scenario.

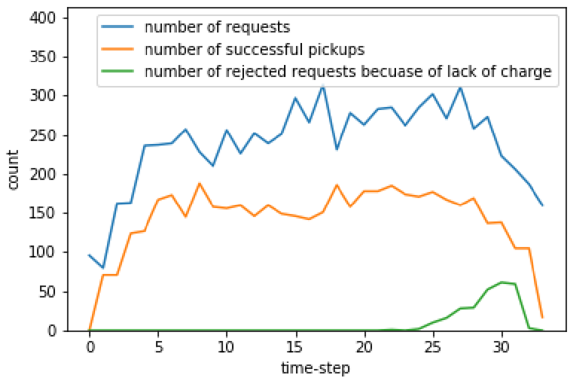


Figure 4.37: Trips (assignment independent of SoC)

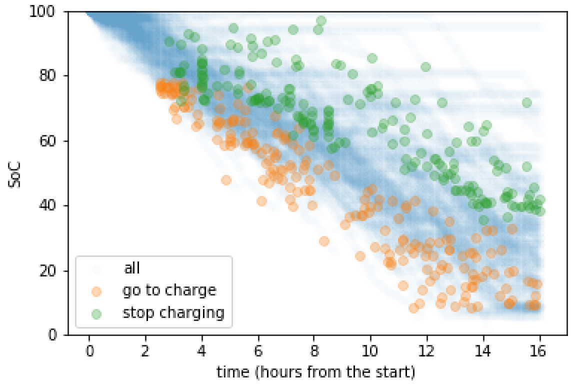


Figure 4.38: SoC of vehicles in time (assignment independent of SoC)

5

Conclusion and recommendation

5.1. Summary of the problem and the proposed solution

An electric-recharging planning algorithm was developed in this study to accompany a dispatching algorithm (*MaaS Dispatcher*) that assigns real-time trip requests to vehicles of a shared taxi fleet owned by an operator. The algorithm decides on when, where, and how much each vehicle should charge, in real-time. It will also relocate idle vehicles if needed. The approach, to designing the algorithm, was to allow maximum flexibility for *MaaS Dispatcher* (not forcing charging on vehicles ahead of time, and restricting their chance of picking up a customer meanwhile), having limited empty routing cost (going to charger and back, and relocating trips), of course while providing enough charge for the expected level of demand to be met. Three sequential mixed linear integer programming (MLIP) optimizations were designed to achieve a pro-active charging planner, that can use aggregated prediction data, run in manageable time, and remain scalable with respect to the fleet-size.

The first level (algorithm $\underline{A}$), only decides on time to charge with steps of 30 minutes and horizon, until the end of the operational day. It will be re-optimized every four hours throughout the day. The second level (algorithm $\underline{B}$) has time-steps of 30 minutes and a horizon of 2.5 hours, and it is re-optimized every four minutes. It will decide on the aggregated level where vehicles of each zone would charge or relocate to. The third level (algorithm $\underline{C}$) is also re-optimized every four minutes, and will decide actions for individual vehicles (e.g. going to a location to charge, stopping to charge and moving to another zone).

5.2. Reflection on the modeling choices

First coherence and stability between the algorithms $\underline{A}$, $\underline{B}$, and $\underline{C}$ are discussed. Following that, is a note regarding the assumption we made earlier that vehicles have enough natural breaks in between their tours to go to charge. Finally, we discuss if we can detect the benefits of a long horizon in algorithm $\underline{B}$ with the current results.

Algorithms $\underline{A}$ and $\underline{B}$ assume that once charging for a vehicle is planned, the vehicle can immediately arrive to the charger, while in reality that takes some time, since the vehicle has to finish its current tour and drive to the charging station. Therefore in simulation, a time lag always appears in catching up with the number of in charge vehicles, planned by algorithm A. This effect is more vivid/consequential when number of in charge vehicles fluctuate. When the number of in charge vehicles stays constant, algorithm $\underline{C}$ can regulate the arrival of new vehicles to chargers to fill in the gaps. Lag happens when going from a small number of vehicles in charge to a high number of vehicles in charge.

In general, the number of vehicles who have to be in charge are not transferable between the time-steps, because of the charger capacity limitation and the demand for vehicles to be active. Not transferable means if the number of vehicles in charge were lower than planned in time-step t, it cannot be compensated by putting more vehicles in charge in time-step $t+1$. However, there is a minimum SoC policy for stopping to charge in algorithm $\underline{B}$, which will keep the vehicles that arrived late to the charger until they have reached a minimum specified SoC (which is derived from algorithm $\underline{A}$), to cope with the mentioned lag in arrival to charging. The same feature also helps to mitigate the effect of underestimating charging demand, so that when vehicles arrive at charger with SoC lower than expected they still charge up to the desired SoC of algorithm $\underline{A}$. It should be noticed that this is not a restriction on charge duration of individual vehicles, so if a vehicle had a higher SoC to begin with, it indeed might end up with an undesirably short charge duration. This policy also produces a lag in following the plan by algorithm $\underline{A}$, when the number of vehicles in charge is decreasing.

The alternative model is to assume a lag between going to charge and arriving at the charger in algorithm B. This version was also tested; it introduced a problem when charging stations were working at capacity. Since algorithm B assumed the new vehicles would not arrive at the charging station until next time-step, it did not move the vehicle in charge out of the chargers, then in algorithm C although there was request to send new vehicles to charge, the request was denied due to charging stations not having free capacity. The policy of enforcing a maximum SoC for vehicles that can continue to charge, somewhat regulated that problem as well. In conclusion, when using algorithm B that assumes immediate arrival of vehicles, we tend to charge some of the vehicles less, just to the minimum SoC allowed for stopping to charge, and in the version where B assumes a lag, we tend to overcharge some of the vehicles, up to the maximum SoC allowed.

As mentioned in the introduction, the *Charging and relocation extension* was designed assuming that there will be enough breakpoints [1] between the trip requests for vehicles to go to charge. This means that we only send vehicles to charge if their current trip finishes before the next time the charging extension is called. This short pre-booking of charging trips allows *MaaS Dispatcher* to have a higher flexibility, in appose to the situation were the charging trip is decided an hour in advance. However, in the base scenario of 8500 trip requests, this condition was not met, in the peak hours. Most vehicles already had a tour lasting longer than the reoptimization time of the *Charging and relocation extension*, thus for a long period, no vehicle was going to the chargers. For the scenarios presented in the result it was decided to force a charging request if the vehicle was becoming free until the next 8 minutes, instead of the 4 minutes originally designed in the algorithm. However, this means that the period $t+4\ min$ to $t+8; min$, is included twice within the decision window of *Charging and relocation extension*. Then the *Charging and relocation extension* has two opportunities to decide, for vehicles whose tour ends between $t+4\ min$ to $t+8; min$, once at time t and once at $t+4$. However, algorithm C's original assumption was that each decision time window is visited once; hence if we do not consider how many vehicles have already been sent to charge, we might end up sending two times more vehicles to charge as originally intended. To fix this, the format of communicating number of vehicles going to charge, from algorithm A to algorithm B, had to change from sending k vehicles to charge in time-step s, to number of vehicles sent to charge from time-step 0 to time-step s must be K, meaning we enforce that total number of vehicles sent to charge until end of time-step s should sum up to K.

In algorithm B, it was decided to prolong the horizon to 5 steps (2.5 hours) with the intention to make a better charging location choice, by considering both inbound and outbound trips from the charger. Alternatively, B could have had only a two-steps horizon. Owing to three obstacles in the scenario design, we cannot observe the benefits of this choice, with the presented results. First, there are only 6 charging stations, that gives a limited choice to algorithm B. Second, as we reach 100% utilization of charging spots, the algorithm is no longer making a choice in the location of charger. Finally, the 8500 requests a day, is near the maximum that a fleet of 150 vehicles can serve, which dampens the benefits of relocation, therefore, even if algorithm B had a choice in the charging location, it would matter less if it was in the direction of the demand or not, as demand is relatively high everywhere.

5.3. Conclusion based on comparison of scenarios

5.3.1. Smart charging Vs. Lazy charging

The results showed that the presented charging algorithm can satisfy around 600 more trips (8% higher acceptance rate) than the lazy charging algorithm, while spending far more on the trips to the chargers. In the smart scenario 7% of all distance traveled is for going to chargers and back, and 5% of all distance is for relocation trips. Lazy charging only invests 0.5% of total distance on the trips to charges. This means with the smart charging operator added 14% to its revenue while increasing the cost of an average trip by 16%. The designed charging algorithm can very well utilize charging stations at close to 100% rate by replacing vehicles in time, without any queuing, unlike the lazy charging who could only use 75% of the slow chargers at the time where more than half of fleet were out of charge.

5.3.2. On effects of assignment extension

Using high SoC vehicles more frequently based on the developed metric increases the number of satisfied requests by 66, which is about one percent of all requests, while increasing the cost of an average trip by 4%. The exact numbers are specific to these scenarios, nonetheless, the improvement from using high SoC

[1] breakpoints are time intervals in which the vehicle has no booked trips, and thus is free

vehicles was limited in all the test scenarios carried out on the side. We can, therefore, conclude that there is not a strong dependence between the charging and the assignment problem.

5.3.3. Behavior of method in presence of demand uncertainty

The first type of uncertainty is about the total requests in the day, which consequently results in an inaccurate prediction of energy consumption through the day. The scenario results showed that with a fixed charging daily plan (from algorithm A) the algorithm can only slightly adjust total charging hours through the day, based on actual average SoC of the fleet. However, the number of trips to the chargers was not adjusted as expected. They even increased in the scenario where realized demand was lower than the expected demand. This is a downside to be addressed in future work.

The second type of uncertainty is regarding the locations of the requests. The operator has an estimate of pickup and drop-off numbers per zone per time-step, which can deviate from the reality. In the results of smart charging, we observed that few vehicles serve less than 10 trips, while an average gap of around 30% remains between satisfied and total requests. This is because the *Charging and relocation extension* is keeping some vehicles in zones were it expects on average 1 or 2 pickups, and failing to appreciate that they would have had a higher chance of a successful pickup if they had been moved to another zone that for example expected 10 pickups but had no vehicles. This is because the algorithm assigns the same value to one unit of pickup in the zone that expects one pickup per time-step and one unit of pickup in the zone that expects ten per time-step. The remedy is having different types of pickups per zone, each having a reward reflecting the confidence in their realization. This way the value of an extra relocation to a zone would decrease gradually, instead of going immediately to zero.

5.4. Theoretical and practical implications

5.4.1. Reflecting on the literature

As analysis of the literature in Section 2.3 shows, few of the studies dealing with dynamic recharging of taxies, consider the location of charge, and the limited number of chargers in them; often they assume that vehicles charge at the closest charging station. The cost of a trip to the charger is not considered either in the daily planning of recharging or the online decision, in majority of the studies. On the other hand, controlling the total number of charging trips throughout the day, and planning under the limited charging capacity, were the main focus in this research.

The method proposed chooses the location of charging according to the likelihood of finding a pickup near the charger. Moreover, unlike most studies, the method is completely flexible regarding the time vehicles stop charging, and will not assume a fixed duration. When to stop charging will be decided online based on availability of other vehicles in the area (for serving passengers), and number of vehicles that have to go to charge. Given this flexibility in duration of charge, this study is one of the first to take to account the drop in charging rate that accrues after 80% SoC.

5.4.2. Applicability outside PTV environment

The applicability of algorithm depends more on the operator's business model than on the accompanying software component such as PTV Dispatcher. The algorithm is designed for the case where the operator owns the taxies, or is responsible for the assignment of requests and recharging the vehicles. In business models such as Uber's for example, the drivers have flexibility in their working hours, they can decline customers, and they choose the place and time they charge their vehicles. The current algorithm would be compatible only if, the operator has reliable aggregate information of working times, and the initial charge the vehicles start with. More importantly, they should have a high compliance with the recommendations of the operator for charging. Taxies with autonomous technology (while controlled centrally) would, for example, align perfectly with the scope of this method.

Concerning chargers, this method is designed for when the operator owns all the chargers. Since the charging needs of vehicles are determined before the start of the day, the operator can also rent out the chargers for extra revenue, in time slots that they are not utilized by its own fleet. The method can be easily extended to include the possibility of charging at public chargers that can be pre-booked. The method is not compatible with public chargers whose availability is not known beforehand, or whose availability cant be guaranteed.

The current PTV dispatcher is myopic, meaning that it will choose a vehicle based on the routing cost, waiting time, and detour, associated with the passengers whose request has already been submitted. However, in shared mobility there is always the potential for smarter dispatching, for example choosing a vehicle with higher seat capacity, for the routes that we expect more passengers on, although have they have not submitted a request yet. In such a case, the charging planner should also be on board with the vision of the dispatching algorithm, by recommending a vehicle who not only has charge for the current request but also the expected (probably longer) tour.

This becomes even more relevant in higher seat capacities, and newly introduced modes such as on-demand public transport, where instead of specific pickup/drop-off points operator dynamically modifies predefined routes. As long as the dispatcher can communicate the expected length of the route, the same method can be used with some modifications. These modifications include for example a higher penalty for going to charge in algorithm $\underline{A}$, since more frequent charging trips, can decrease the probability of sharing trips, and the ability of vehicles to take on longer tours. The *charging and relocation extension* also has to book charging trips longer ahead of time (e.g. changing the current 4 minutes to 30 minutes). For more traditional modes such as buses with fixed routes, the model is not directly applicable anymore, as we would require an integrated decision on time, and location of charging due to the fixed terminals, and timetables.

5.5. Limitations and recommendations

5.5.1. Tuning parameters

The advantage of the developed algorithm is that the operator has the option to balance the number of requests that it likes to satisfy and the cost it is willing to pay for executing the requests. It also provides the operator with the ability to reschedule charging, and respond to known disruptions such as chargers out of service or a day with exceptional low temperature, which would otherwise disrupt operation significantly.

The disadvantage is that while tuning various parameters of the model provide us with options, exploiting these options are not easy. Even with a simulator that can well replicate the real-world operation, getting the parameters that give the best outcome is burdensome. Besides, parameters are not a one fit all configuration. They will depend on fees, demand level, etc.

Algorithm $\underline{A}$ has the most grip on the costs, it will set the total number of trips to the chargers. The total number of trips to the chargers, and percent utilization of the chargers are good predictors of the average cost of the trips to chargers. The more vehicles you have to select per time-step for sending to charge, the more restricted the choice, and the more distance they have to drive to get to the charger. That is why the operator should start with tuning algorithm $\underline{A}$. The tuning will depend on fees of the operator, its dispatching algorithm, and consequently the profit it can collect per operating hour. The tuning of algorithm $\underline{A}$ should be straight forward for the operator; it needs to set the relative profit of the vehicle from working for one full-time-step, to the cost of an average trip to charger. It does not require a deep understanding of the charging algorithm, and the balance can be achieved after a few operation days.

In algorithms $\underline{B}$ and $\underline{C}$ operator tunes to what degree to follow algorithm $\underline{A}$, based on the the real-time and higher resolution information that is available to them. The order of the cost variation caused by tuning $\underline{B}$ and $\underline{C}$ is much lower than $\underline{A}$, however, the tuning requires a basic understanding of the algorithms, and more iterations, under diverse scenarios. The first information required for tuning is the added benefit of a relocation depending on the likelihood of a passenger in origin compared to the potential destination. The second category required is loss from not following number of in charge vehicles planned by algorithm $\underline{A}$ and loss from not following number going to charge vehicles planned by algorithm $\underline{A}$.

5.5.2. Capacity of the model to be expanded

The main theoretical disadvantage is that the method is not easily generalizable, all parts all tightly integrated, and changing the assumption of one, would mean having to reconstruct the rest. Introducing flexibilities such as drivers taking a break at their desired time, and charging at public chargers, or stochasticity in travel time and electricity prices would disturb the core assumptions of the method. Moreover, the current algorithm assumes a homogeneous fleet, however, it can be expanded to include gasoline cars, hybrids, and different battery capacities, and consumption rates. In that case, all calculations and penalties based on SoC have to be translated to remaining driving range, to be comparable across different types of vehicles. With that said, while details of the model are prone to be obsolete with demand to integrated more or different features into the model, integrating aggregated longer-term information on the immediate decision of time, place, and duration to charge, is nevertheless valuable to experiment with, and more likely to survive the test of time.

5.5.3. Consumption function

The plans of the smart charging algorithm can only be as good as the charging prediction that it uses. The decharging curve used in this study was only a function of distance and its prediction was assumed to be perfect, which is not a good representation of the reality. For the algorithm to be operational it needs to be accompanied by a range prediction algorithm according to the required accuracy for each step of the charging planner, which was discussed in Section 3.5.

As stated earlier algorithm $\underline{A}$ has the most influence on the charging plan, and its requirement regarding the consumption prediction is modest. In algorithm $\underline{A}$ operator mainly uses the historical data of energy consumption as an input; it only needs to adjust for time-varying variables in the consumption function on aggregate level, such as a drop in temperature.

In real-time, the charging planner should prevent vehicles from running out of charge, the most challenging part of consummation prediction is then getting a good upper bound on consummation of vehicles in the *assignment extension*. This is where the operator would probably need to perform a link based (in oppose to route based) calculation. Ideally we would like to pre-calculate the less dynamic components of the consummation function [2], however most of these effects are interdependent [3] such as temperature and speed, and pre-calculating overall combination is often not efficient. The charging planner should be paired with one of many studies such as [6] that offer varying accuracies under the allowable computation time.

[2] e.g. gradient is always constant, whereas temperature and speed (or speed profile) are more dynamic
[3] are not additive or multiplicative

Bibliography

[1] Lina Al-Kanj, Juliana Nascimento, and Warren B. Powell. Approximate Dynamic Programming for Planning a Ride-Sharing System using Autonomous Fleets of Electric Vehicles. 10 2018. URL http://arxiv.org/abs/1810.08124.

[2] Benedict Evans. cars-and-second-order-consequences, 2017. URL https://www.ben-evans.com/benedictevans/2017/3/20/cars-and-second-order-consequences.

[3] Tao Chen, Bowen Zhang, Hajir Pourbabak, Abdollah Kavousi-Fard, and Wencong Su. Optimal Routing and Charging of an Electric Vehicle Fleet for High-Efficiency Dynamic Transit Systems. *IEEE Transactions on Smart Grid*, 9(4):3563–3572, 7 2018. ISSN 1949-3053. doi: 10.1109/TSG.2016.2635025. URL https://ieeexplore.ieee.org/document/7763884/.

[4] Jacques Desrosiers and Marco E. Lübbecke. A primer in column generation. In *Column Generation*, pages 1–32. Springer US, 2005. ISBN 0387254854. doi: 10.1007/0-387-25486-2{_}1.

[5] John F. Franco, Marcos J. Rider, and Ruben Romero. An MILP model for the plug-in electric vehicle charging coordination problem in electrical distribution systems. In *2014 IEEE PES General Meeting | Conference & Exposition*, pages 1–5. IEEE, 7 2014. ISBN 978-1-4799-6415-4. doi: 10.1109/PESGM.2014.6939006. URL http://ieeexplore.ieee.org/document/6939006/.

[6] Stefan Grubwinkler, Tobias Brunner, and Markus Lienkamp. Range Prediction for EVs via Crowd-Sourcing. In *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6. IEEE, 10 2014. ISBN 978-1-4799-6783-4. doi: 10.1109/VPPC.2014.7007121. URL http://ieeexplore.ieee.org/document/7007121/.

[7] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28 (1):100, 1979. ISSN 00359254. doi: 10.2307/2346830.

[8] Riccardo Iacobucci, Benjamin McLellan, and Tetsuo Tezuka. Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid. *Transportation Research Part C: Emerging Technologies*, 100:34–52, 3 2019. ISSN 0968-090X. doi: 10.1016/J.TRC.2019.01.011. URL https://www.sciencedirect.com/science/article/pii/S0968090X18309197.

[9] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. Planned Route Optimization For Real-Time Vehicle Routing. In *Dynamic Fleet Management*, pages 1–18. Springer US, Boston, MA, 2007. doi: 10.1007/978-0-387-71722-7{_}1. URL http://link.springer.com/10.1007/978-0-387-71722-7_1.

[10] Peraphan Jittrapirom, Valeria Caiati, Anna-Maria Feneri, Shima Ebrahimigharehbaghi, María J Alonso-González, Jishnu Narayan, M J Alonsogonzalez@tudelft Nl, J N Sreekantannair@tudelft, (J N Nl, Soora Rasouli, Harry Timmermans, and Dujuan Yang. Mobility as a Service: A Critical Review of Definitions, Assessments of Schemes, and Key Challenges. 2(2):13–25, 2017. doi: 10.17645/up.v2i2.931. URL http://hdl.handle.net/2066/174112.

[11] Jaeyoung Jung, R Jayakrishnan, and Keechoo Choi. Shared-Taxi Operations with Electric Vehicles. Technical report, 2012. URL http://www.its.uci.edu.

[12] Jun-Li Lu, Mi-Yen Yeh, Yu-Ching Hsu, Shun-Neng Yang, Chai-Hien Gan, and Ming-Syan Chen. Operating electric taxi fleets: A new dispatching strategy with charging plans. In *2012 IEEE International Electric Vehicle Conference*, pages 1–8. IEEE, 3 2012. ISBN 978-1-4673-1561-6. doi: 10.1109/IEVC.2012.6183233. URL http://ieeexplore.ieee.org/document/6183233/.

[13] & Moura S. J. Nicolas, L. *Optimal Routing and Charging of Electric Ride-Pooling Vehicles in Urban Networks*. PhD thesis, UC Berkeley: Energy, Controls, and Applications Lab, 2016. URL https://escholarship.org/uc/item/4m5369gs.

[14] Warren B. Powell. Dynamic Models of Transportation Operations. *Handbooks in Operations Research and Management Science*, 11:677–756, 1 2003. ISSN 0927-0507. doi: 10.1016/S0927-0507(03)11013-4. URL https://www.sciencedirect.com/science/article/pii/S0927050703110134.

[15] Harilaos N Psaraftis, Min Wen, and Christos A Kontovas. Dynamic Vehicle Routing Problems: Three Decades and Counting. 67(1):3–31, 2016. doi: 10.1002/net.21628. URL https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.21628.

[16] Jakob Puchinger, Peter J Stuckey, Mark Wallace, and Sebastian Brand. Dantzig-Wolfe Decomposition and Branch-and-Price Solving in G12. Technical report. URL https://people.eng.unimelb.edu.au/pstuckey/papers/colgen.pdf.

[17] Ons Sassi, W. Ramdane Cherif-Khettaf, and Ammar Oulamara. Multi-start Iterated Local Search for the Mixed Fleet Vehicle Routing Problem with Heterogenous Electric Vehicles. pages 138–149. Springer, Cham, 2015. doi: 10.1007/978-3-319-16468-7{_}12. URL http://link.springer.com/10.1007/978-3-319-16468-7_12.

[18] Michael Schneider, Andreas Stenger, and Dominik Goeke. The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, 48(4):500–520, 11 2014. ISSN 0041-1655. doi: 10.1287/trsc.2013.0490. URL http://pubsonline.informs.org/doi/abs/10.1287/trsc.2013.0490.

[19] Chien Ming Tseng, Sid Chi Kin Chau, and Xue Liu. Improving Viability of Electric Taxis by Taxi Service Strategy Optimization: A Big Data Study of New York City, 2018. ISSN 15249050.

[20] Guang Wang, Xiuyuan Chen, Fan Zhang, Yang Wang, and Desheng Zhang. Experience: Understanding Long-Term Evolving Patterns of Shared Electric Vehicle Networks. 12 2018. URL http://arxiv.org/abs/1812.07499.

[21] Guang Wang, Xiaoyang Xie, Fan Zhang, Yunhuai Liu, and Desheng Zhang. bCharge: Data-Driven Real-Time Charging Scheduling for Large-Scale Electric Bus Fleets. In *2018 IEEE Real-Time Systems Symposium (RTSS)*, pages 45–55. IEEE, 12 2018. ISBN 978-1-5386-7908-1. doi: 10.1109/RTSS.2018.00015. URL https://ieeexplore.ieee.org/document/8603191/.

[22] Zaiyue Yang, Lihao Sun, Jiming Chen, Qinmin Yang, Xi Chen, and Kai Xing. Profit maximization for plug-in electric taxi with uncertain future electricity prices. *IEEE Transactions on Power Systems*, 2014. ISSN 08858950. doi: 10.1109/TPWRS.2014.2311120.

[23] Zaiyue Yang, Tianci Guo, Pengcheng You, Yunhe Hou, and S. Joe Qin. Distributed Approach for Temporal-Spatial Charging Coordination of Plug-in Electric Taxi Fleet. *IEEE Transactions on Industrial Informatics*, pages 1–1, 2018. ISSN 1551-3203. doi: 10.1109/TII.2018.2879515. URL https://ieeexplore.ieee.org/document/8521690/.

[24] Zaiyue Yang, Tianci Guo, Pengcheng You, Student Member, Yunhe Hou, Senior Member, and S Joe Qin. Distributed Approach for Temporal-Spatial Charging Coordination of Plug-in Electric Taxi Fleet NOMENCLATURE Parameters and variables for temporal scheduling. 2018. doi: 10.1109/TII.2018.2879515. URL http://www.ieee.org/publications_standards/publications/rights/index.html.

[25] Ming Zhu, Xiao Yang Liu, and Xiaodong Wang. Joint transportation and charging scheduling in public vehicle systems - A game theoretic approach. *IEEE Transactions on Intelligent Transportation Systems*, 2018. ISSN 15249050. doi: 10.1109/TITS.2018.2817484.