Dynamic planning for simultaneous recharging and relocation of shared electric taxies
A sequential MILP approach

Jamshidi, Helia ; Correia , Gonçalo H.A.; van Essen, J. Theresia; Nökel, Klaus

DOI
10.1016/j.trc.2020.102933

Publication date
2021

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Citation (APA)
Jamshidi, H., Correia , G. H. A., van Essen, J. T., & Nökel, K. (2021). Dynamic planning for simultaneous recharging and relocation of shared electric taxies: A sequential MILP approach. *Transportation Research Part C: Emerging Technologies*, *125*, 1-27. Article 102933. https://doi.org/10.1016/j.trc.2020.102933

Transportation Research Part C 125 (2021) 102933

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Dynamic planning for simultaneous recharging and relocation of shared electric taxies: A sequential MILP approach

Helia Jamshidi [a], Gonçalo H.A. Correia [b,*], J. Theresia van Essen [c], Klaus Nökel [d]

[a] PTV Group, Haid-und-Neu-Strasse 15, 76131 Karlsruhe, Germany
[b] Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Transport & Planning, P.O. Box 5031, 2600 GA Delft, the Netherlands
[c] Delft University of Technology, Delft Institute of Applied Mathematics, P.O. Box 5031, 2600 GA Delft, the Netherlands
[d] PTV Group, Haid-und-Neu-Strasse 15, 76131 Karlsruhe, Germany

ARTICLE INFO

Keywords:
EV
Ride-sharing
Recharging
Relocation
Optimization
MILP

ABSTRACT

Short driving range, limited chargers, and long charging times challenge the profitability of electric taxi operations. In this paper, a charging algorithm is developed to accompany a taxi service with online trip requests, which uses a private charging infrastructure for both slow and fast charging. The vehicle charging curves are assumed to be piece-wise linear functions. The proposed algorithm uses historical operation data to generate a pro-active planning that avoids queuing of vehicles. The algorithm is built upon three sequential, iterative, finite-horizon Mixed Integer Linear Programs. This iterative process, in which the three MILPs are solved sequentially, allows the current time-step to be optimized, while taking future time-steps into account. This is achieved by optimizing over multiple time-steps, but only implementing the current time-step in each iteration. The sequential aspect of the algorithm allows the vast amount of information over time and space to be exploited for charging trip decisions in real time, while maintaining a tractable computation time. The first level with the longest horizon is an aggregated, daily problem, that plans the charging duration required for the fleet. The second level has a horizon of up-to three hours and is an aggregated, zone-based problem for determining charger selection and empty vehicle relocations. The third level translates the outputs of the first two problems to executable decisions for individual vehicles based on their real-time location, state of charge, and assigned passengers. The first level is the most computationally expensive and is solved using Column Generation. The performance of the first two levels is then independent of the fleet size, which makes the algorithm highly scalable. A case study with travel data for the city of Barcelona is used to test the model. Results show that the proposed method can utilize the full capacity of the charging infrastructure, and improve the number of accepted requests by 14% compared to employing a naive charging rule.

1. Introduction

Electrification of mobility services offers additional environmental benefits resulting from a decrease in local pollutant emissions and a higher energy use efficiency of the electric motors. Operational cost is also lower, because of cheaper energy, lower energy

* Corresponding author.
E-mail address: g.correia@tudelft.nl (G.H.A. Correia).

https://doi.org/10.1016/j.trc.2020.102933
Received 3 June 2020; Received in revised form 9 December 2020; Accepted 15 December 2020
Available online 8 March 2021
0968-090X/

expenditure per kilometer (as vehicles can restore charge while driving downhill), lower repair costs, and longer lifetime (Benedict Evans, 2017). Despite the environmental benefits of a transition towards zero-emission vehicles, the diffusion of electric vehicles (EVs) faces barriers such as range limitations, a limited number of 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 air conditioning. While private owners can rely on overnight charging for their daily use, for taxi drivers this is not viable due to the great power consumption. Therefore, they have to charge at least once during the day.

A long term study on charging patterns among E-taxis (ET) in Wang et al. (2019), which has the largest number of E-taxis (ETs) in the world, 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 conclude that charging stations play a key role in large-scale ET promotion and even if enough charging points are deployed, the unbalanced temporal or spatial charging demand and supply reduce the efficiency of the overall charging network.

While the problem of optimally locating public charging stations and wireless charging facilities, in urban environments, have been dealt with rigorously in He et al. (2015), Li et al. (2016), Dong et al. (2014), Chen et al. (2017), Tu et al. (2016), Riemann et al. (2015), and Birrell et al. (2015), the experience in Wang et al. (2019) suggests that more efficient utilization of the charging infrastructure, and smooth operation of E-taxis, requires a software back-end that tracks vehicles, their State of Charge (SoC) and customer ride requests to manage the electric fleets. The charging planner should navigate a taxi driver to the best charging location based on the availability of chargers, remaining battery range, and overall demand for charging during the day.

In this paper, we develop a method to decide on the charging operations in real time. We consider a system defined by an operator who owns a fleet of ETs and earns revenue from satisfying requests that come in real-time from the clients. It is assumed that passengers request the service knowing their ride might be shared, and subject to a detour. The number of vehicles that can be charged at any moment is constrained by both the number of vehicles and the number of available chargers. The algorithm is designed to accompany and extend the functionality of dispatching algorithms that can assign vehicles to requests with no battery constraints (e.g. fossil fuel cars), to include EV operations. Therefore, two decision processes have to be developed to ensure the smooth operation of the fleet. The first is sending vehicles to charge at a specific charging station with a specified duration. The second is to prevent the dispatching algorithm from choosing low SoC vehicles to serve trips.

We assume that the operator only uses private charging stations, hence, there is no competition with other fleets for the chargers. The chargers offer either slow charging or fast charging. Electricity prices are variable throughout the day but are known beforehand. More restrictive scenario parameters include homogeneity of the fleet in terms of fuel type, capacity, consumption rate, driving range, and charging compatibility with existing charging systems. The charging curve is a piecewise linear function with a lower charging rate after SoC of 80%. The size of the fleet is fixed throughout the day, which means that we are assuming that all taxis are active on the network even during off-peak times. Finally, the consumption rate is assumed to be a function of only distance for the case study.

The paper is organized as follows. In the next section, a literature review is presented where the scientific contribution of this paper is identified. This is followed by Section 3 in which the proposed methodology is introduced, including the three-level optimization for charging and relocation. In Section 4, we apply the model to a case study in Barcelona and compare the results from the proposed method with results obtained using a basic charging rule. This section also presents the performance of the models. In Section 5, the main conclusions and future work are shown.

2. Related work

Studies tackling the recharging problem for ET fleets have different sets of assumptions, leading to a varying range of complexities. 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 marginal profit associated with adding the first vehicle in an area is the same as the tenth or the fiftieth vehicle in that area (Al-Kanj et al., 2020; Tseng et al., 2018; Yang et al., 2018). The methodological consequence is that the fleet does not need to be controlled centrally, and therefore, a global optimal solution can be achieved by letting vehicles choose the best action for themselves. In this study, the problem is assumed to be constrained by the existing requests in the different areas, and therefore, marginal utility of a new vehicle diminishes as the number of vehicles grows at some locations. Consequently, the algorithm developed in this paper would be unnecessarily complex if the fleet is much smaller compared to the available demand. Nevertheless, this is not expected in a real setting where supply and demand are close to an equilibrium. Related to this, the work of Yang et al. (2018) has several advantages and disadvantages over our work. The solution approach of Yang et al. (2018) allows for stochastic future electricity prices while we only allow for variable but known electricity prices (Table 1). In addition, while we use an optimization approach to maximize the operators benefit (i.e. aim for a system optimum), Yang et al., 2018 solves the assignment of drivers to chargers by obtaining a Nash Equilibrium which reflects a free choice for drivers (i.e. user equilibrium). Which of these approaches is best, depends on the business model of the operator, e.g. whether drivers are paid per hour or per customer. The game theoretical approach used in Yang et al. (2018) may be better suited if resources are distributed by government or union and not managed by a system operator as is the case in our work. The advantage of our work is that 1) it takes into account that operation areas can be saturated by taxies (diminishing return for more vehicles in the same area), 2) it makes a long term plan of when to charge by taking into account the total charger capacity and total charging demand of the system in one day, and 3) it assigns a charging station not only based on the current queue time but also on future demand in the proximity of the charging station.

The other impacting assumption is whether or not the number of chargers in the considered region is limited. Surprisingly, only a few studies address this issue. The implication of unlimited chargers is that it leads to less binding constraints on the relation between vehicles' decisions, hence, the impact of a charging vehicle will not propagate in time as much as is the case when there are charging limitations. This will also change the horizon required to solve the 'when to charge' problem optimally. An overview of related studies is presented in Table 1.

In the set of studies without constrained demand, (Tseng et al., 2018) has solved the problem for one vehicle with a Markov Decision Process (MDP). The state of the vehicle was defined by its location (nearest junction), SoC (discrete levels), and time (with one minute time-steps). The actions available for drivers to choose from were waiting, relocating and charging. Using New York taxi data for internal combustion vehicles, probabilities for successful passenger pickups were pre-calculated for all junctions and time-steps. In addition, the probability of a passenger commuting from the origin junction to the destination junction was calculated. The transition function (for definition and further detail on MDP please refer to Bellman, 1957) was adapted for ETs by modifying the transition probabilities based on the vehicles' SoC. More specifically, if the destination was out of reach with a certain battery charge, the probability was set to zero. The first limitation of using this method is, as mentioned, that actions of one vehicle do not affect the actions of others. The second limitation is that the method indeed needs an exceptionally rich data set to estimate such a detailed transition function without generalization.

In a recent paper (Al-Kanj et al., 2020), which considers unconstrained demand for a ride-sharing system, approximate dynamic programming (ADP) was compared to a myopic policy in a joint car-rider allocation, car relocation, and recharging problem. In the dynamic programming approach, future trip demand was taken into account under a fixed demand scenario. The value function was defined over discrete states that capture time, location and state of charge, and is assumed to be linear with the number of cars. The value of a single car in the value function literally means that if a car is dropped in a zone a_1 with state of charge 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 approximately be equal to this value. The approach of Al-Kanj et al. (2020) is appropriate when dealing with stochastic demand and when no posterior information on demand is available. However, it cannot account for higher average demand, since the value function cannot use demand as an input. Furthermore, in this implementation of ADP, the authors do not consider the problem of where to charge as well as the charger capacity restrictions, as displayed in Table 1. We aim to fill this gap by developing solution approaches that operators can readily use in practice. Furthermore, our solution approach for the charging problem improves upon the approach provided in Al-Kanj et al. (2020) by offering the possibility to include the prediction of demand as a variable and by considering ride-hailing demand as finite in each operational zone.

In the set of studies with constrained demand, in an autonomous and electric car-sharing context, Iacobucci et al. (2019) 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 long time scales to minimize both approximate waiting times and electricity costs (30 min time steps and horizon of 5 h). Routing and relocation are optimized at shorter time scales to minimize waiting times (2 min time steps and horizon of 30 min). There are two main reasons why real life ride-hailing operators cannot employ this method in the current state. As reported by the authors, the computational complexity of the problem grows more than linearly with the number of vehicles and number of nodes, which makes using this approach impracticable for more than a few tens of vehicles. Second, as seen in Table 1, they assume that stations have an unlimited charger capacity, while charger capacity plays a significant role in shaping our method. Furthermore, it is assumed that unsatisfied demand in one time-step propagates to the next time-step and will not be rejected, while we consider a strict maximum waiting time of a client in this study.

In Zhu et al. (2018), where the authors use an offline procedure, the requests of one day ahead are simulated without considering charging to get the energy that the fleet needs throughout the day. Then, the time to charge each vehicle is optimized while taking into account variable electricity prices. In the online optimization, EVs are grouped together. Then, each group has to decide what percentage of vehicles it will allocate to transporting clients, and consequentially, which vehicles will be assigned to charge. In order to make this decision, a utility function is defined for each group where the number of in-charge EVs is equal to the number planned by the offline charging planner. The online problem is solved with a cake cutting game. In the cake cutting game, each of the vehicle groups gets a piece of transportation and charging cake, based on how much they value the piece. The main limitation is that they do not associate a cost to the trips of EVs going to charging stations. There is no limit on the number of chargers, as they are assumed immediately available, thus, the problem of where to charge is not addressed. 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.

In conclusion, Markov decision processes are best equipped to incorporate uncertainty in operation and demand through the use of transition functions and value functions (Al-Kanj et al., 2020; Tseng et al., 2018). However, they fall short in modeling constrained demand and different demand scenarios, where subsequent information is available compared to a base scenario for which they were optimized for. The main take from Iacobucci et al. (2019) and Zhu et al. (2018) is the need to plan charging over a long horizon. Iacobucci et al. (2019) uses larger time steps for planning the time and location of a charging operation, while keeping it at an individual vehicle level. Zhu et al. (2018) only plans the number of vehicles that charge in each time step, and does not select charging locations or individual vehicles. The latter is preferred, as planning for time and location of charging five hours ahead, under demand stochasticity, raises computational complexity without providing benefits. However, in the model used in Zhu et al. (2018) for the offline planning of the time to charge, the authors assumed that all vehicles can use the same shared mega-battery. While that would make the problem easier to solve, it implies 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 highly 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 mean that 10 trips can be served in both cases. The real operation with limited chargers is closer to the latter case, and thus, at any given time, we must have more reserved

Table 1
Methods in Electric Vehicle Charging Problem.

		This study}	Franco et al. (2014)}	Lu et al. (2012)}	Jung et al. (2012)}	Sassi et al. (2015)}	Chen et al. (2018)}	Schneider et al. (2014)}	Nicolas and Moura (2016)}	Wang et al. (2018)}	Yang et al. (2018)}	Iacobucci et al. (2019)}	Al-Kanj et al. (2020)}	Tseng et al. (2018)}	Zhu et al. (2018)}
Problem	Diminishing marginal utility of vehicles	✓				✓		✓	✓	✓		✓			✓
	Limited no. chargers	✓	✓	✓			✓				✓				
	Dynamic requests	✓		✓	✓						✓	✓	✓	✓	✓
	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[d]	PDP[e]	VRP[d]	PDP[e]	E-bus	E-taxi	Carsharing	Ridesharing	NY taxi	E-taxi
Method	Optimization	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Method	MIP[f]	MIP[f]	Heuristics	Heuristics	MIP[f]	MIQCP[g]	MIP[f]	MPC[h]	MDP[f]	thresholding + GT	MPC[h]	DP[i]	MDP[j]	Cake cutting game
	Method sub class					local search	VNS[a] /TS[b]	Clustering/ VNS[a]/ TS[b]/ CG[c]		AVI[k]	VI[l]				
Objective	Energy cost	✓	✓			✓	✓	✓		✓	✓	✓	✓		✓
	Revenue	✓				✓	✓			✓	✓	✓	✓	✓	✓
	Waiting time of passengers						✓					✓			

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

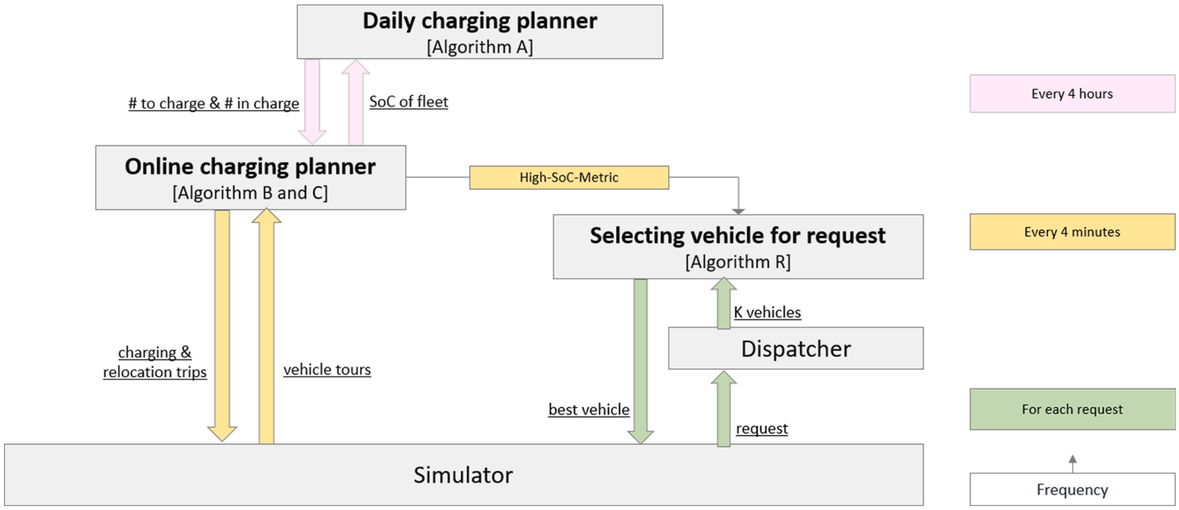


Fig. 1. Algorithmic approach.

charge in the system than if we were to assume that there is a shared battery.

The proposed algorithm in this study addresses many shortcomings of the literature on operating ETs with dynamic requests. These shortcomings include scalability with respect to fleet size while acknowledging saturation of demand, using aggregate demand prediction information, complying with charger capacity restrictions, allowing for flexible charging duration, and modeling a drop in charging rate after obtaining a SoC of 80%. The algorithm considers the cost of driving to chargers both in determining the frequency of charging trips and chargers' location. Moreover, it uses the SoC of vehicles while assigning vehicles to requests and exploits the dependency between the relocation problem and the charging location problem. While we do simplify stochasticity, we do not leave out any subproblems and major operational restrictions (e.g. charger capacity, scalability), and therefore, our solution approach can immediately be used in practice.

3. Methodology

3.1. Outline of the proposed algorithm

The problem with naive charging strategies is that they do not consider the energy demand for the entire operation time, leaving vehicles uncharged at the end of the operation time. By acting passively, the operator would either not have enough chargers, or not have enough time to reboot the fleets' energy up to the desired amount. In a Smart Charging algorithm, optimizing charging time and duration must take into consideration the energy demand for the whole day. However, the information available over the required horizon is only reliable in an aggregated form, such as the total energy required by the fleet as opposed to the energy consumption of each vehicle. By reliable, we mean that there is a more systematic pattern than there is randomness in the behavior of the demand for charging. Therefore, for time to charge and duration of charge, we need to make aggregated decisions over a long horizon. The decision

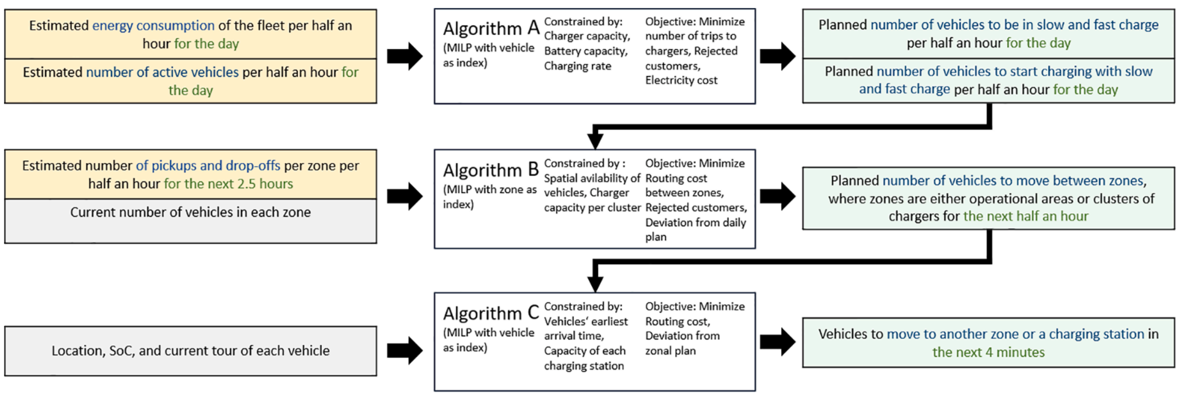


Fig. 2. Flow of information among the three algorithms.

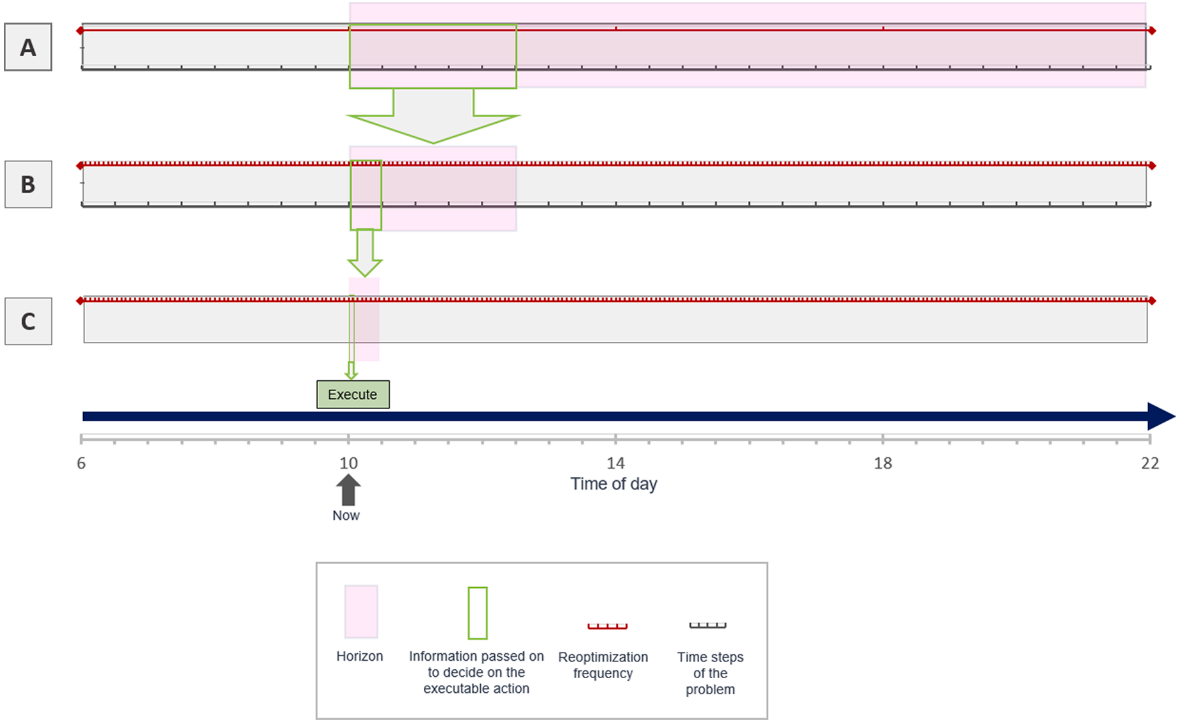


Fig. 3. Time-line of the algorithms A, B, and C.

should include the number of vehicles that should be charging throughout the day as well as their average charging duration in the same period. In this paper, the component that deals with time and duration of charge over the course of the whole day is called *Daily charging planner* and it will be designated as algorithm A, which is introduced in Section 3.2. The *Daily charging planner* guides the *Online charging planner* that decides in real time which vehicle has to charge.

The *Online charging planner*, where the decision of where to charge is taken, has a two to three hour horizon. This is because the problem of finding a charging location is not only influenced by the inbound trip to the charging station but also the outbound trip, meaning where the vehicle should move next after it is done charging. Once the horizon extends beyond the charging time of vehicles, the algorithm can prioritize charging stations that are located in zones where customers are to be expected. Historical data on the number of pick ups per zone is used to estimate where the trip requests are likely to be originated and is only available in an aggregated form; the algorithm cannot know in advance where each vehicle would get a customer. In other words, the average charging time, for which we like to plan thereafter, is longer than an average trip duration, over which the operator has full information on the passengers and tours of individual vehicles. This constraint on the information available can be addressed by dividing the decision into two steps. In the first step, aggregated decisions are made on how many vehicles to move from a zone to charging zones or other zones; this results in flows of vehicles between zones. In the second step, the operator selects the vehicles to satisfy this required flow between zones for the horizon over which the operator has information on the location of vehicles. Therefore, the *Online charging planner* actually consists of two optimization algorithms, namely algorithms B and C, which are introduced in Section 3.3.

The relation between all mentioned algorithms is illustrated in Fig. 1. The simulator represents the real world. At the beginning of the day, it will connect to the *Online charging planner*, which will request an aggregate schedule for charging from the *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. From there, the simulator will start processing requests one by one. Each time a new request comes in, the dispatcher checks if the request can be accepted, and will send k vehicles that can service the request to *Selecting vehicle for request* (Algorithm R). Then, algorithm R will return the best available vehicle based on the vehicles' SoC. When four minutes have passed in the simulation, the simulator will call the *Online charging planner* again. Fig. 2 outlines in more detail how information is communicated between the algorithms. Yellow in Fig. 2 means that the algorithm is consuming a forecast, gray boxes are information about the system in real time, and green represents the output we use from the solution of the algorithm. In the middle column, a high level summary of the most prominent constraints and objectives of the problems are included.

Charging and re-positioning trips are the result of a three level hierarchical optimization. The first level only decides on when to charge with steps of 30 min and a horizon that goes up to the end of the operational day. This first level is re-optimized every four hours throughout the day. The second level has steps of 30 min and a horizon of two to three hours, and is re-optimized every four minutes. It will decide on an aggregated level where vehicles from each zone should charge or relocate to. The third level's horizon is as long as the planned duration of the tours of the vehicles and is re-optimized every four minutes. For all levels, only the decisions that have to start before the next re-optimization are executed (see Fig. 3). The time-steps specified in Fig. 3 for algorithms A and B show the discretization of time used in their respective formulations.

3.2. Daily charging planner (Algorithm A)

3.2.1. Modeling

A Mixed Integer Linear Program (MILP) is developed to optimize the charging of the fleet during a day. At this level of modeling, each vehicle is reduced to what takes place in its battery. It does not matter where the vehicles are located, or which specific trips they would be serving. In this model, vehicles 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 min.

The charging curve of a vehicle is a piecewise linear function with a lower charging rate above SoC of 80%. In order to impose a different charging rate above 80% SoC, we model the battery of a taxi as two fictitious batteries, one with a capacity of 80% and another with a capacity of 20%. The second one can only be charged when the first one is full, and the first one can only decharge if the second one is empty. Demand for active taxis is given as input to the problem, where the more demand the fleet can satisfy the better. Vehicles pay for charging and are penalized every time they start to charge, which is imposed to prevent a plan with frequent and short charging periods. The price of charging is defined per KW gained, and is relative to the price of overnight charging in the depot (which is the minimum). In reality, the total SoC should be spread out over many idle vehicles, since we need charged vehicles available throughout the operational area. Therefore, the number of charged vehicles needed at each time step is larger than the number of active vehicles. A parameter is introduced to express the overall desired number of charged taxis for each time-step. Satisfying this

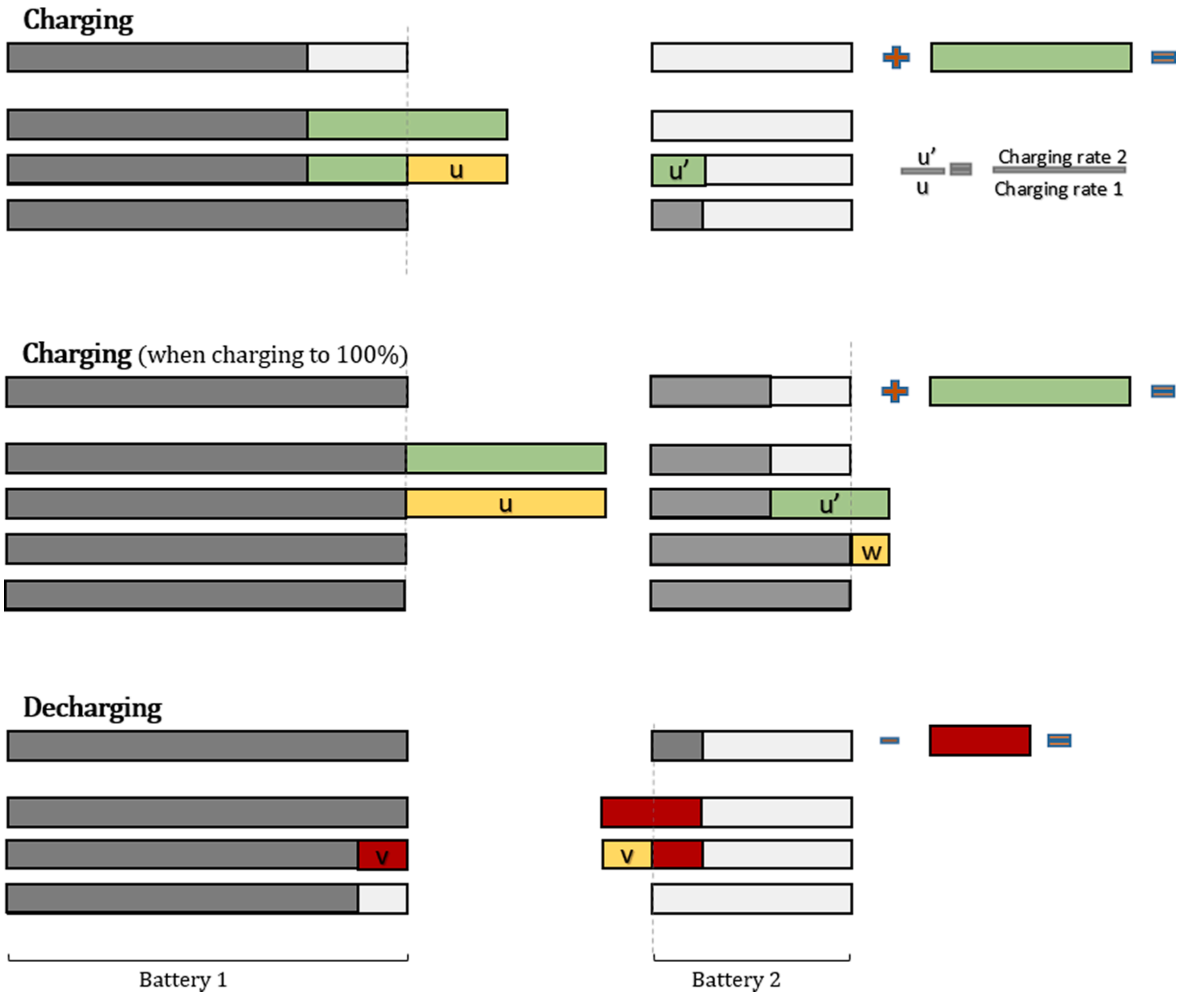


Fig. 4. Charging and decharging in fictitious batteries. The dark gray is the charged portion of the battery, the green is the charge we intend to add in one time-step, according to the charging rate of the first fictitious battery (the higher rate), and in red is the amount of battery that is consumed in one time-step. The value that variables u_{itc}, w_{it}, and v_{it} take is shown with the length of the yellow color bar with symbol **u**, **w**, and **v**, respectively. The value of **u'** is obtained by multiplying u by $\frac{\textit{charging rate from 80\% to 100\%}}{\textit{charging rate from 0\% to 80\%}}$ to account for the lower charging rate above 80%. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

number is rewarded in the optimization model.

3.2.2. Mathematical formulation

The set of vehicles and set of time-slots are given by sets I and T, respectively. For each time-slot $t \in T$, we have the predicted number of required active vehicles D_t and the number of desired charged vehicles B_t (with SoC higher than 20%). Each vehicle $i \in I$ has a given initial charge L_{i0}, which can be split into the initial charge of the first fictitious battery $L^1_{i0} = \min(L_{i0}, 0.8)$ and of the second fictitious battery $L^2_{i0} = \max(L_{i0} - 0.8, 0)$. The charge of the battery of vehicle $i \in I$ at time-slot $t \in T$ is given by L^1_{it} and L^2_{it} for the first and second fictitious battery, respectively.

In addition, we consider two types of charging given by set $C = \{slow, fast\}$. The charging rate of charging type $c \in C$ up to SoC of 80% is given by r^1_c in fraction per time-step. The charging rate of charging type $c \in C$ after SoC of 80% is given by r^2_c. The average energy consumption rate per time step is given by e (in fraction per time-step). This coefficient can vary on different days based on the weather and temperature.

For each vehicle $i \in I$ at time-slot $t \in T$ for charging type $c \in C$, we have binary variables s_{itc}, f_{itc}, and d_{it}. Binary variables s_{itc} are 1 when vehicle $i \in I$ starts charging at a charger of type $c \in C$ at time-slot $t \in T$, and 0 otherwise. Similarly, binary variables f_{itc} are 1 when vehicle $i \in I$ stops charging at a charging station of type $c \in C$ at time-slot $t \in T$, and 0 otherwise. Lastly, binary variables d_{it} are 1 when vehicle $i \in I$ is serving demand at time-slot $t \in T$.

Constraints (1) impose for each time-slot $t \in T$ that the total number of vehicles serving demand cannot be more than the number of required active vehicles D_t.

$$\sum_{i \in I} d_{it} \leqslant D_t, \ \forall t \in T \tag{1}$$

In addition, Constraints (2) reduce the size of the solution space by limiting the number of visits to a charging station to M for each vehicle $i \in I$.

$$\sum_{c \in C} \sum_{t \in T} s_{itc} \leqslant M, \ \forall i \in I \tag{2}$$

We introduce binary variables y_{itc} which are 1 when vehicle $i \in I$ is charging at a charger of type $c \in C$ at time-slot $t \in T$, and 0 otherwise. The following constraints establish the relation between y_{itc}, s_{itc}, and f_{itc}.

$$y_{itc} = \sum_{k=1}^{t} (s_{ikc} - f_{ikc}), \ \forall i \in I, t \in T, c \in C \tag{3}$$

Note that Constraints (3) also ensure that a vehicle cannot stop charging before it starts charging or start charging when it is already charging, since variables y_{itc} can only attain values 0 and 1.

Having defined variables y_{itc}, the following constraints impose that the total number of vehicles in-charge at time-slot $t \in T$ is bounded by the available number of chargers N_c of type $c \in C$.

$$\sum_{i \in I} y_{itc} \leqslant N_c, \forall t \in T, \forall c \in C \tag{4}$$

Next, we explain how we determine the values of variables L^1_{it} and L^2_{it}, which represent the SoC of the battery of vehicle $i \in I$ at time $t \in T$ for the first and second fictitious battery, respectively. First of all, we limit the maximum value of L^1_{it} and L^2_{it} to 0.2 and 0.8, respectively.

$$0 \leqslant L^1_{it} \leqslant 0.8, \forall i \in I, \forall t \in T \tag{5a}$$

$$0 \leqslant L^2_{it} \leqslant 0.2, \forall i \in I, \forall t \in T \tag{5b}$$

However, when charging vehicle $i \in I$ during time $t \in T$ with charging type $c \in C$, it can happen that these bounds are exceeded. This excess charge for the first and second battery are captured by variables u_{itc} and w_{it}, respectively (see also Fig. 4). The excess charge of the first battery is actually charge of the second battery, and thus, should be added to L^2_{it}. In order to determine this excess charge u_{itc} of the first battery for vehicle $i \in I$ at time $t \in T$ resulting from charge type $c \in C$, we first introduce variable L^{1+}_{it} which represents the total charge of the first battery after charging during time-step $t \in T$.

$$L^{1+}_{it} = L^1_{i(t-1)} + \sum_{c \in C} r^1_c \cdot y_{itc}, \ \forall i \in I, \forall t \in T, c \in C \tag{6}$$

Now, binary variables a_{itc} are one when the charge of the first battery L^{1+}_{it} of vehicle $i \in I$ at time $t \in T$ after charging at a charger of type $c \in C$ is more than 0.8, and zero otherwise. To ensure that a_{itc} can only be one when $L^{1+}_{it} \geqslant 0.8$ and when the vehicle is charging at a charger of type $c \in C$, i.e., $y_{itc} = 1$, we introduce the following constraints.

$$a_{itc} \leqslant 0.2 + L^{1+}_{it}, \forall i \in I, \forall t \in T, \forall c \in C \tag{7}$$

$$a_{itc} \leqslant y_{itc}, \forall i \in I, \forall t \in T, \forall c \in C \tag{8}$$

Even though Constraints (7) and (8) do not force a_{itc} to become 1 when $L_{it}^{1+} \geqslant 0.8$, Constraints (5a) forces a_{itc} to be 1 in this case. When a_{itc} is 1, u_{itc} should represent the excess charge of the battery, therefore, $u_{itc} = L_{it}^{1+} - 0.8$. When $a_{itc} = 0$, u_{itc} should be equal to 0. This is ensured by the following constraints.

$$u_{itc} \leqslant L_{it}^{1+} - 0.8a_{itc}, \forall i \in I, \forall t \in T, c \in C \tag{9}$$

$$u_{itc} \leqslant a_{itc}, \forall i \in I, \forall t \in T, c \in C \tag{10}$$

Something similar happens when the second battery is decharged. Namely, when decharging, the SoC of the second battery might drop below zero which means that the first battery has to be decharged as well. In order to model this, variables L_{it}^{2-} are introduced which represent the charge of the second battery of vehicle $i \in I$ at time $t \in T$, where L_{it}^{2-} is allowed to take negative values whereas L_{it}^{2} is not. Recall that binary variable d_{it} is one when vehicle $i \in I$ is serving demand at time $t \in T$.

$$L_{it}^{2-} = L_{i(t-1)}^{2} - e \cdot d_{it}, \forall i \in I, \forall t \in T \tag{11}$$

Binary variables b_{it} are only allowed to take value one when L_{it}^{2-} is negative for vehicle $i \in I$ at time $t \in T$ and when vehicle $i \in I$ was serving demand at time $t \in T$. This is ensured by the following two constraints.

$$b_{it} \leqslant 1 - L_{it}^{2-}, \forall i \in I, \forall t \in T \tag{12}$$

$$b_{it} \leqslant d_{it}, \forall i \in I, \forall t \in T \tag{13}$$

Even though Constraints (12) and (13) do not force b_{it} to become 1 when L_{it}^{2-} is negative, Constraints (5b) forces b_{it} to be 1 in this case. The amount by which the second battery is decharged in excess is then represented by variables v_{it} for vehicle $i \in I$ at time $t \in T$ (see also Fig. 4). This is the amount of charge that should be used from the first battery. Note that v_{it} should be zero when the second battery is not decharged in excess, i.e., when b_{it} is zero.

$$v_{it} \leqslant -L_{it}^{2-} + 0.8(1 - b_{it}), \forall i \in I, \forall t \in T \tag{14}$$

$$v_{it} \leqslant b_{it}, \forall i \in I, t \in T \tag{15}$$

Now we have everything in place to determine the true SoC of both fictitious batteries. The initial SoC of both batteries of vehicle $i \in I$ are given by L_{i0}^{1} and L_{i0}^{2}, respectively. The next equation calculates the correct value of L_{it}^{1} for vehicle $i \in I$ at time $t \in T$ where u_{itc} is the excess charge of the first battery resulting from charging type $c \in C$, which thus should be subtracted, and v_{it} is the excess decharge of the second battery that should be subtracted from the first battery.

$$L_{it}^{1} = \sum_{k=0}^{t} \left(\sum_{c \in C} \left(r_c^1 \cdot y_{ikc} - u_{ikc} \right) - v_{ik} \right) + L_{i0}^{1}, \forall i \in I, t \in T \tag{16}$$

To properly calculate the SoC L_{it}^{2} of the second battery of vehicle $i \in I$ at time $t \in T$, we should add the excess charge u_{itc} resulting from charging type $c \in C$ of the first battery to the second battery, while taking into account the different charging rates, and add the amount of charge v_{it} that has been decharged in excess from the second battery. In addition, variable w_{it} represents the charge over 100% (see also Fig. 4).

$$L_{it}^{2} = \sum_{k=0}^{t} \left(\sum_{c \in C} \frac{r_c^2}{r_c^1} \cdot u_{ikc} - e \cdot d_{ik} + v_{ik} - w_{ik} \right) + L_{i0}^{2}, \forall i \in I, t \in T \tag{17}$$

As mentioned before, algorithm A incentivizes having some number of reserved charged taxis scattered through the operational area. We keep track of charged vehicles by introducing binary variable q_{it} which can only attain value one if the SoC of vehicle $i \in I$ is higher than 20%, and it is not busy at time $t \in T$. Therefore, constraints (18) guarantee that q_{it} equals 0 if the SoC of vehicle $i \in I$ falls below 20% at time $t \in T$ and constraints (19) ensure that vehicles cannot be active and go to charging at the same time.

$$q_{it} \leqslant L_{i(t-1)}^{1} + 0.8, \forall i \in I, \forall t \in T \tag{18}$$

$$d_{it} + q_{it} + \sum_{c \in C} y_{itc} \leqslant 1, \forall i \in I, \forall t \in T \tag{19}$$

Finally, constraints (20) impose that the total number of charged vehicles plus the slack variable Q_t for the number of required charged vehicles at time $t \in T$ must be more than the number of required charged vehicles, B_t.

$$\sum_{i \in I} q_{it} + \sum_{i \in I} d_{it} + Q_t \geqslant B_t, \forall t \in T \tag{20}$$

The objective is to minimize the following function: cost of going to charge plus charging cost minus reward from active vehicles plus cost of not having enough charged taxis. The operator needs to have estimates on the average revenue β obtained from 30 min of a vehicle's operation along with the average cost α_c of going to a charging station of type $c \in C$. To find the best time to charge, the operator also needs electricity prices throughout the day, introduced as the penalty p_{tc} for the cost of charge gained with charging type $c \in C$ and proportional to the additional electricity price at time $t \in T$ in relation to the overnight price. γ is the penalty for the slack number of charged taxis.

$$\min \sum_{c\in C}\left(\alpha_c \sum_{t\in T}\sum_{i\in I} s_{itc} + \sum_{t\in T} p_{tc}\left(\sum_{i\in I} r_c^1 \cdot y_{itc} - \sum_{i\in I} u_{itc} + \sum_{i\in I}\frac{r_c^2}{r_c^1}\cdot u_{itc} - \sum_{i\in I} w_{it}\right)\right) - \beta \sum_{t\in T}\sum_{i\in I} d_{it} + \gamma \sum_{t\in T} Q_t \tag{21}$$

3.2.3. *Solving the problem with Column Generation*

In the beginning of the day, when 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 exactly the same. Column Generation (CG) (see Dantzig and Wolfe, 1960) can take advantage of this symmetry. Dantzig-Wolfe decomposition is used to divide the original problem into a subproblem that generates daily plans for individual vehicles given their battery constraints whilst the master problem decides how many vehicles should follow each of those plans. The master problem makes sure that global constraints (associated with more than one vehicle) such as limited charger capacity, available demand and the desired number of charged vehicles are satisfied.

CG iteratively solves the master and subproblem. At each iteration of column generation, the dual multipliers obtained by solving the linear relaxation of the master problem are used to generate daily plans that improve the objective function of the relaxed master problem. We terminate the CG algorithm if there is no significant improvement of the objective function of the relaxed master problem over the last 15 iterations. Since columns (daily plans from the subproblem) generated in earlier stages have lower quality, they will stop appearing as basic variables in the master problem after some time. Therefore, to trim these redundant columns, every column that has not been used by the master problem for the past 15 iterations is eliminated. After the last iteration, we solve the integer master problem given the generated set of columns. The resulting solution is not proven optimal and the gap cannot be quantified. The reason for this is twofold. First, because the new columns (individual vehicle plans) are generated based on the relaxed master problem, and therefore, columns that may improve the integer master problem are not known; second, because the process is terminated once the improvement of the objective function value slows down, and not when no new column is available. One may still compare the obtained objective function value 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 400 vehicles, the vehicles are clustered into groups of 40 vehicles where within each group vehicles act identically to each other. In this example, we would have 10 daily plans for individual vehicles to start the CG algorithm with. Algorithm 1 describes the iterative procedure. In pactice, the operator may keep a pool of most used daily plans over several days and use that as the initial columns. This in turn reduces the number of iterations required and drastically shortens solving time.

Algorithm 1.

Column Generation

Initialize a set of daily plans (columns) for individual vehicles by sub-optimally solving an aggregated version of the original problem
while True **do**
Solve the linear relaxation of the master problem
if improvement of the objective function value of the relaxed master problem over the past 15 iterations is less than 0.5% **then**
Terminate loop
Get dual multipliers from the relaxed master problem
Solve the subproblem to generate new daily plans (columns)
Add the generated columns to the master problem
if column is not used in the optimal relaxed solution of the master problem for the past 15 iterations **then**
Eliminate column
Solve the integer master problem with the generated set of columns

As stated, the subproblem generates daily plans for individual vehicles. Such a daily plan for an individual vehicle is given by variables y_{tc}, d_t, and q_t introduced in Section 3.2.2, which describe whether the vehicle is charging with charging type $c \in C$ at time $t \in T$, is serving demand at time $t \in T$, or is idle with a battery charge higher than 20%, respectively. The set K of daily plans is given as input to the master problem where variables y_{tc}, d_t, and q_t are represented by parameters $\widetilde{y}_{ktc}$, $\widetilde{d}_{kt}$, and $\widetilde{q}_{kt}$ for each daily plan $k \in K$. The main decision variable in the master problem is now the number of vehicles x_k that follow daily plan $k \in K$. The cost C_k of daily plan $k \in K$ is given by Eq. (29). The constraints that a daily plan should fulfill are given by Constraints (2) and (3), (5a)-(5b), and (7)–(19). Therefore, these constraints are included in the subproblem while Constraints (1) on serving demand, Constraints (4) on available number of chargers, and Constraints (20) on total number of charged vehicles are included in the following relaxed master problem.

$$\min\sum_{k\in K}C_kx_k+\gamma\sum_{t\in T}Q_t \tag{22}$$

$$\sum_{k\in K}x_k\widetilde{y}_{ktc}\leqslant N_c,\forall t\in T,c\in C \tag{23}$$

$$\sum_{k\in K}x_k\widetilde{d}_{kt}\leqslant D_t,\forall t\in T \tag{24}$$

$$\sum_{k\in K}x_k\widetilde{q}_{kt}+\sum_{k\in K}x_k\widetilde{d}_{kt}+Q_t\geqslant B_t,\forall t\in T \tag{25}$$

$$x_k\geqslant 0,\forall k\in K \tag{26}$$

$$Q_t\geqslant 0,\forall t\in T \tag{27}$$

As stated before, the constraints that a daily plan should fulfill are given by Constraints (2) and (3), (5a)-(5b), and (7)–(19). Therefore, these constraints are included in the subproblem. However, when solving the subproblem, we create one daily plan for a single vehicle. This means that set I contains a single element, and therefore, index $i\in I$ can be removed from all variables and constraints in (2) and (3) and (5a)-(5b), (7)–(19).

The objective function of the subproblem is now given by the reduced cost of the daily plan. Here, y^{π}_{tc}, d^{π}_{t}, and q^{π}_{t} are the dual variables of Constraints (23)–(25), respectively.

$$\min\sum_{c\in C}\left(\alpha_c\sum_{t\in T}s_{tc}+\sum_{t\in T}p_{tc}\left(r^1_c\cdot y_{tc}-u_{tc}+\frac{r^2_c}{r^1_c}\cdot u_{tc}-w_t\right)\right)-\beta\sum_{t\in T}d_t$$
$$-\sum_{t\in T}\sum_{c\in C}y_{tc}y^{\pi}_{tc}-\sum_{t\in T}d_td^{\pi}_t+\sum_{t\in T}q_tq^{\pi}_t+\sum_{t\in T}d_tq^{\pi}_t \tag{28}$$

Eq. (29) gives the cost of a generated daily plan, which is input for the master problem.

$$C=\sum_{c\in C}\left(\alpha_c\sum_{t\in T}s_{tc}+\sum_{t\in T}p_{tc}\left(r^1_c\cdot y_{tc}-u_{tc}+\frac{r^2_c}{r^1_c}\cdot u_{tc}-w_t\right)\right)-\beta\sum_{t\in T}d_t \tag{29}$$

3.3. Online charging planner

3.3.1. Outline

As previously introduced, the online planner needs to optimize on an aggregated level so that it can use the aggregated pickup and drop-off information available, over a two hour horizon. The aggregated decisions correspond to determining empty vehicle flows between zones over which data on pickup and drop-offs are available. Next to the operation zones, there are charging zones which contain a bundle of charging stations close together. The charging zones do not need to be within the boundaries of the operation zones, since they are independent. The output of the aggregated level optimization (algorithm B) is then the number of vehicles moving between operation zones, or traveling from operation zones to charging zones and vice versa.

After obtaining the results of algorithm B, the operator knows that it is best, for example, to re-position k free vehicles from operation zone o to operation zone d and send q free vehicles from operation zone o to charging zone m, throughout the next 30 min. Knowing the time and location of the vehicles' last passenger drop-offs, the operator can choose the best vehicles to fulfill the recommended flows by algorithm B. This is a task for optimization algorithm C. Algorithm C aims at minimizing the routing cost of going to charge and relocating. It makes sure that the charger utilization rate follows the plan laid out by algorithm B, that the vehicles with lowest SoC are chosen for going to charge, and that in-charge vehicles stop charging at the planned SoC (corresponding to the solution from algorithm A).

3.3.2. Algorithm B

As for algorithm A, the set of time-slots and charging types are given by T and C, respectively, where time-slots now refer to steps in the horizon of algorithm B (e.g. 5 min in the case study in Section 4). For each charging type $c\in C$ and time-step $t\in T$, algorithm B aims to satisfy the number of vehicles that have to be in-charge, Y_{tc}, along with the number of vehicles that have to start charging, X_{tc}. This is input obtained from algorithm A. The latter is required to distribute charge among vehicles, otherwise algorithm B would keep the same vehicles in-charge to avoid the cost of routing vehicles in and out of the chargers.

The charging stations owned by the operator are clustered geographically in charging zones. The set of charging zones for charging type $c\in C$ is denoted by E_c with $\cup_{c\in C}E_c=E$. The set of operation zones is denoted by Z. At $t=1$, there are v_{o1} vehicles present in zone $o\in Z\cup E$. The main decision variables are non-negative integer variables x_{odt} which represent the number of vehicles relocating between operation zones $o\in Z$ and $d\in Z$ at time $t\in T$, going from operation zone $o\in Z$ to charging zone $d\in E$ at time $t\in T$, and moving back to operation zone $d\in Z$ from charging zone $o\in E$ at time $t\in T$.

Algorithm B also tries to relocate vehicles to zones where demand is expected. Thus, we need as input the predicted number of pickups, p_{ot}, along with the predicted number of drop-offs, d_{ot}, for each operation zone $o \in Z$ at time $t \in T$. While predicting the number of pickups and drop-offs, the operator should also account for the rate of shared rides.

Not all predicted pickups can be satisfied in each time step, given the real-time state of the fleet. Therefore, we introduce non-negative integer variables p^*_{ot} as the number of successful pickups from operation zone $o \in Z$ at time $t \in T$ and non-negative integer variables s_{ot} as the slack number of pickups.

$$p^*_{ot} = p_{ot} - s_{ot}, \forall t \in T, \ \forall o \in Z \tag{30}$$

We then have to impose that the conservation of vehicles holds by enforcing that the number of vehicles who leave an operation zone at a certain time step must be less than or equal to the number of vehicles currently in the zone. Therefore, Constraints (31) impose that the number of vehicles in operation zone $o \in Z$ should be non-negative at time $t \in T$, when all vehicles with origin $o \in Z$ have left from the zone, but no vehicle with destination $o \in Z$ has arrived to the zone. The arriving vehicles are not counted because that would allow infinite trips between a pair of zones at each time step. This number is calculated in Constraints (31) as the initial number of vehicles v_{o1} in operation zone $o \in Z$ minus the number of vehicles that left the zone to relocate, charge, or serve a trip from time step 1 until time step t, plus the number of vehicles that have arrived to zone $o \in Z$ until time step $t-1$.

$$v_{o1} - \sum_{k=1}^{t}\left(p^*_{ok} + \sum_{d \in Z \cup E} x_{odk}\right) + \sum_{k=1}^{t-1}\left(d_{ok}\lambda_k + \sum_{d \in Z \cup E} x_{dok}\right) \geqslant 0, \forall t \in T, \forall o \in Z \tag{31}$$

Furthermore, in each time-step, the total number of pickups and drop-offs should be the same to prevent counting vehicles twice. The number of drop-offs d_{ot} predicted for zone $o \in Z$ at time $t \in T$ is therefore discounted by λ_t, which is calculated in Eq. (32) as the ratio of the total number of successful pickups $\sum_{o \in Z} p^*_{ot}$ at time $t \in T$ and the total number of predicted pickups $\sum_{o \in Z} p_{ot}$ at time $t \in T$.

$$\lambda_t = \frac{\sum_{o \in Z} p^*_{ot}}{\sum_{o \in Z} p_{ot}}, \forall t \in T \tag{32}$$

Similarly, Constraints (33) ensure that the number of vehicles in charging zone $o \in E$ should be non-negative and less than or equal to the charger capacity m_o of charging zone $o \in E$ for each time step $t \in T$.

$$0 \leqslant v_{o1} - \sum_{k=1}^{t}\sum_{d \in Z} x_{odk} + \sum_{k=1}^{t}\sum_{d \in Z} x_{dok} \leqslant m_o, \forall t \in T, \forall o \in E \tag{33}$$

Constraints (34) impose that the number of vehicles in-charge for charging type $c \in C$ at time $t \in T$ must equal Y_{tc} as planned by algorithm A. We introduce non-negative integer variables B'_{tc} as the surplus number of vehicles in-charge at $t \in T$ and charging type $c \in C$, which are used to relax the constraints in two cases. The first case arises when vehicles are on their way to a charging station of charging type $c \in C$, while algorithm B already counts them as in-charge. The number of in-charge vehicles is then allowed to be higher than what has been planned by algorithm A given by Y_{tc}. The second case is when we have more charge consumption throughout the day than initially predicted, and vehicles take longer to boot up to the desired SoC level. In this case, we also need to allow the number of in-charge vehicles to be more than Y_{tc}. We introduce non-negative integer variables B_{tc} as the slack variables for the number of vehicles in-charge for charging type $c \in C$ at time $t \in T$, in the case that there are not enough vehicles to send to charge.

$$\sum_{o \in E_c} v_{o1} + \sum_{k=1}^{t}\sum_{o \in Z}\sum_{d \in E_c} x_{odk} - \sum_{k=1}^{t}\sum_{o \in E_c}\sum_{d \in Z} x_{odk} + B_{tc} = Y_{tc} + B'_{tc}, \forall t \in T, \forall c \in C \tag{34}$$

Constraints (35) make sure that the number of vehicles going to charge is greater than or equal to X_{tc}. These constraints bound the average duration of charge from above. The reason that we allow this to be greater than X_{tc} is that algorithm B can cut the charging of vehicles in zones where vehicles are needed and instead add an additional charging trip in another zone to avoid an expensive relocation. Non-negative integer variables G_{ct} represent the slack number of vehicles that have to start charging with type $c \in C$ at $t \in T$ for when there are not enough vehicles to send to charge.

$$\sum_{o \in Z}\sum_{d \in E_c} x_{odt} + G_{ct} \geqslant X_{tc}, \forall t \in T, \forall c \in C \tag{35}$$

Since algorithm B manages the aggregated vehicle flows, it cannot directly access the SoC of each vehicle. However, some information on the SoC level of vehicles is required to obtain practical solutions from algorithm B. For example, algorithm B should not force highly charged vehicles to start charging, should not force newly arrived vehicles at a charging station to leave, and should not let vehicles with no battery charge relocate or serve customers. To apply these restrictions, we use the number of vehicles with a SoC below or above a certain SoC threshold to communicate aggregated SoC information of the fleet with algorithm B.

While the average duration of charge for all vehicles is bounded by Constraints (35), algorithm B may still charge vehicles in some charging zones over 100%. Therefore, we need to impose a minimum number of outbound trips from individual charging zones to make sure that fully charged vehicles leave the charging stations. Constraints (36) impose that the total number of vehicles leaving charging zone $o \in E$ until time $t \in T$ must be greater than or equal to the number of vehicles v^{2+}_{ot} of which the SoC reaches L^{2+}_t. Here, L^{2+}_t represents the upper bound SoC for being in-charge at time $t \in T$. The value of L^{2+}_t is equal to $\min(A^f_t + 0.25, 1)$ with A^f_t being the average SoC of vehicles stopping charging in algorithm A at time $t \in T$. Note that the value of v_{ot}^{2+} is derived from the solution obtained by algorithm A and that it is an input for algorithm B. In theory, vehicles could be moving back and forth between operation and charging zones, and thus, increase the total number of vehicles leaving a charging zone. However, this behaviour is discouraged, since it is penalized by the routing cost in the objective function. In addition, Constraints (39) enforce vehicles to stay in a charging zone for a minimum charging duration.

$$\sum_{k=1}^{t} \sum_{d \in Z} x_{odk} \geqslant \sum_{k=1}^{t} v_{ok}^{2+}, \forall t \in T, \forall o \in E \tag{36}$$

Similarly, we need to prevent sending vehicles with a high SoC, for example 90%, to charging zones in the first time-step. We set a SoC upper bound L_t^{2-} for vehicles that can be send to charge. The value of L_t^{2-} is given by $\min(A_t^s + 0.15, 1)$ where A_t^s is the average SoC of vehicles going to charge in algorithm A at time $t \in T$. Constraints (37) enforce that the number of vehicles that are sent to charge from operation zone $o \in Z$ at time $t = 1$ is smaller than or equal to the number of vehicles v_o^{2-} with a SoC lower than L_1^{2-}. We do not enforce this condition for $t > 1$ since algorithm B cannot track the movement of individual vehicles. This can be problematic if the charged vehicles are not evenly distributed over the zones. However, since we only execute the output for $t = 1$, this will not be a problem when we apply the algorithm in practice. Note again that the value of v_o^{2-} is an input, which is derived from the solution obtained from algorithm A.

$$\sum_{d \in E} x_{od1} \leqslant v_o^{2-}, \forall o \in Z \tag{37}$$

Similar limitations apply for vehicles that are assigned to relocate or serve customers. Vehicles that do not have enough battery for any trip should not be relocated to places where we expect demand. This is only enforced for the first time-step. Constraints (38) impose that the number of vehicles assigned to serve customers or relocate from zone $o \in Z$ cannot be greater than the number of vehicles v_o^+ in zone $o \in Z$ that have enough charge for at least one trip at time $t = 1$. Also here the value of v_o^+ is an input, which is derived from the solution obtained from algorithm A.

$$\sum_{d \in Z} x_{od1} + p_{o1}^{*} \leqslant v_o^+, \forall o \in Z \tag{38}$$

Finally, we need to ensure that vehicles have reached a lower bound SoC L_t^- when they stop charging at time $t \in T$. Without this restriction, the solution obtained by algorithm B might force a vehicle that just arrived to a charging station to leave immediately; or to plan a short charging duration for charging stations that are more centrally located, and a long charging duration for stations farther away from the city center. The value of this lower bound SoC L_t^- is given by $\max(A_t^f - 0.25, 0.35)$. From the solution obtained by algorithm A, we can determine the number of vehicles v_{ot}^- in charging zone $o \in E$ that were put to charge before $t = 1$ and will reach L_t^- at time $t \in T$. Constraints (39) state that the total number of vehicles that can leave charging zone $o \in E_c$ at time $t \in T$ is bounded from above by v_{ot}^- plus the vehicles that are assigned to go charging with type $c \in C$ in charging zone $o \in E_c$ at least MCD_c time-steps ago. Here, MCD_c denotes the minimum charging duration for charging type $c \in C$.

$$\sum_{k=1}^{t} \sum_{d \in Z} x_{odk} \leqslant \sum_{k=1}^{t} v_{ok}^- + \sum_{d \in Z} \sum_{k=1}^{t-MCD_c} x_{dok}, \forall t \in T, \forall o \in E_c \forall c \in C \tag{39}$$

The objective of algorithm B is to minimize all routing costs in addition to the penalty for unsatisfied pickups and vehicles that were not able to charge. The average cost of going from one zone to the other is given by e_{od} for $o, d \in Z \cup E$. The penalty for an unsatisfied pickup is α and acts as an upper bound on charge spent on a relocation trip. β_c is the penalty for the unsatisfied number of in-charge vehicles for charging type $c \in C$ and θ_c is the penalty for the unsatisfied number of going to charge vehicles for charging type $c \in C$. β' is the penalty for the number of surplus vehicles in-charge.

$$\min \sum_{t \in T} \left(\sum_{o,d \in Z} e_{od} x_{odt} + \sum_{o \in Z, d \in E} e_{od} x_{odt} + \sum_{o \in E, d \in Z} e_{od} x_{odt} + \sum_{o \in Z} \alpha s_{ot} + \sum_{c \in C} \left(\beta_c B_{tc} + \theta_c G_{tc} + \beta' B'_{tc} \right) \right) \tag{40}$$

3.3.3. Algorithm C

Given the aggregated vehicle flows from algorithm B, algorithm C selects individual vehicles to relocate, to go to charge or to stop charging. In addition, different from algorithm B, vehicles are assigned to charging stations within a charging zone and not to charging zones. The aim of algorithm C is to choose the low charged vehicles that are close to charging stations as much as possible and at the same time reach the targeted charger utilization rate (determined by algorithm B), throughout the next 30 min. In terms of repositioning, algorithm C should select charged vehicles that are close to the destination zone. The going-to-charge and relocating assignments take place after the last planned drop-off of the vehicles. For vehicles that are currently in-charge, the algorithm provides four evenly distributed options over the next 30 min, which is the time-step of algorithm B, for the vehicle to stop charging, namely in 0, 7.5, 15, or 22.5 min.

First, the potential vehicles that can satisfy flows obtained from algorithm B should be selected. Potential in this context means that a vehicle has its last drop-off at the latest 30 min from the current time instant. Algorithm C can only direct a vehicle to a charging station if the vehicle is classified as chargeable, i.e., has a SoC lower than a threshold derived from algorithm A. If the vehicle is directed to an operation zone, the vehicle should have an adequate charge for a relocation trip plus the charge required for an average trip with passengers. To assign vehicles to a destination zone or charging station, we need the consumption of all vehicle-destination pairs. If the SoC of a vehicle is lower than the consumption needed for a certain destination, then the vehicle cannot be assigned to this particular destination zone or charging station. With that said, there is always a reachable charging station available for all vehicles, since this is guaranteed by algorithm R (see Section 3.4). The elimination based on SoC might only filter charging stations that are farther away than the closest charging station. For each vehicle-destination pair, the SoC of the vehicle at the last drop-off, the estimated time of arrival (ETA) at the potential destination zone, and estimated consumption to the potential destination zone is passed on as input. Further inputs include the current SoC of each vehicle, along with the ETA of vehicles on their way to charging stations, and charging trips that were previously decided on but have not yet arrived. Naturally, the flows and the total number of in-charge vehicles are passed down from algorithm B.

As for algorithms A and B, the set of vehicles is denoted by set I and the set of charging types by set C. Set Z represents the set of operation zones and E_c is the set of charging zones of charging type $c \in C$ with $\cup_{c \in C} E_c = E$. Each charging zone $e \in E$ is a set in itself containing the charging stations within this zone. The set of charging stations of charging type $c \in C$ is denoted by S_c with $\cup_{c \in C} S_c = S$. Note that each charging station $s \in S$ is an element of exactly one charging zone $e \in E$, i.e., $\forall s \in S, \exists! e \in E$ with $s \in e$. The main decision variables are binary variables x_{id}, which equal 1 when vehicle $i \in I$ is directed to operation zone $d \in Z$ or charger $d \in S$ after its last drop-off and 0 otherwise, and binary variables x_{idt}, which equal 1 when vehicle $i \in I$ that is in-charge has to move to operation zone $d \in Z$ at time $t \in T$ and 0 otherwise.

The input from algorithm B is given by the charging flow B^c_{od} from operation zone $o \in Z$ to charging zone $d \in E$, relocation flow B^r_{od} with $o, d \in Z$, flow B^f_{od} from charging zone $o \in E$ to operation zone $d \in Z$, and the total number of vehicles B^i_c planned to be in-charge with charging type $c \in C$.

The charging flow B^c_{od} represents the number of vehicles that should go from operation zone $o \in Z$ to charging zone $d \in E$ over the next 30 min. However, in algorithm C, we plan the charging trips exactly at the time the vehicle drops off its passenger(s). If the majority of drop-offs accrue in a time window much smaller than 30 min, then the algorithm would be forcing the vehicles that were planned to start charging over a period of 30 min, to start charging over a much shorter time period, and hence, overestimate the number of vehicles going to charge. In this case, we reduce the going-to-charge flow relative to the average expected start time m of the charging trips of all vehicles that are available for going to charge. Therefore, we use $B'^c_{od} = \min\left(1, \frac{m}{n}\right) B^c_{od}$ with n being half the time step duration of algorithm B (i.e. 15 min).

Constraints (41)–(43) ensure that algorithm C satisfies the relocation and charging flows given by algorithm B. Here, L_i represents the last drop-off location $L_i \in Z$ of vehicle $i \in I$ or the charging station $L_i \in S$ vehicle $i \in I$ is at.

For the relocation flows with origin $o \in Z$ and destination $d \in Z$, we introduce non-negative slack variables s^r_{od} which put a bound on the maximum combined penalty of charge spend on a relocation trip and the SoC of the vehicle.

$$\sum_{i \in I | L_i = o} x_{id} + s^r_{od} = B^r_{od}, \forall o, d \in Z \tag{41}$$

For the charging flows with origin $o \in Z$ and destination $d \in S$, we introduce non-negative slack variables s^c_{od} which put a bound on the maximum combined cost of charge spend on a trip going to charge and the penalty for the SoC of the vehicle. Note that the charging flow resulting from algorithm B has as destination a charging zone $e \in E$, whereas the vehicles are assigned to a specific charging station $d \in e$ by algorithm C.

$$\sum_{i \in I | L_i = o} \sum_{d \in e} x_{id} + s^c_{oe} = B'^c_{oe}, \forall o \in Z, e \in E \tag{42}$$

Constraints (43) ensure that we satisfy the flows B^f_{od} from charging zone $o \in E$ to operation zone $d \in Z$.

$$\sum_{i \in I | L_i \in o} \sum_{t \in T} x_{idt} \geqslant B^f_{od}, \forall o \in E, \forall d \in Z \tag{43}$$

To track at which time in the next 30 min vehicles enter and exit the charging stations, the time-step from algorithm B has been divided into 4 time-steps in algorithm C. Since we know the last drop-off location of vehicle $i \in I$ and the time needed to arrive at charging station $s \in S$, we know at which time vehicle $i \in I$ will arrive at charging station $s \in S$ when it is assigned to this charging station. This information is captured by parameters a_{ist} which are 1 when vehicle $i \in I$ arrives at charging station $s \in S$ before time-step $t \in T$ and 0 otherwise. This allows introducing Constraints (44) which impose that the number of vehicles in-charge with charging type $c \in C$ should be equal to B^i_c for each time step $t \in T$. The slack and surplus number of in-charge vehicles with charging type $c \in C$ at time $t \in T$ are represented by non-negative integer variables s^i_{tc} and s'^i_{tc}, respectively. The surplus in-charge variable allows vehicles to reach the desired SoC before leaving the charging station in the days that energy consumption was underestimated. In addition, we have the number of planned trips pc^i_{tc} to charging stations of charging type $c \in C$ that arrive before $t \in T$ and the number of vehicles ic_c that were initially in-charge with charging type $c \in C$.

$$\sum_{k=0}^{t}\left(\sum_{i\in I|L_i\in Z}\sum_{d\in S}x_{id}a_{idk}-\sum_{i\in I|L_i\in S}\sum_{d\in Z}x_{idk}\right)+ic_c+pc^i_{tc}=B^i_c-s^i_{tc}+s'^i_{tc},\forall t\in T,\forall c\in C \tag{44}$$

Constraints (45) ensure that the capacity cp_s of charging station $s\in S$ is not violated, and thus, avoid queuing in the charging stations. From previous runs of algorithm C, we obtain the number of planned trips ps_{st} to charging station $s\in S$ that arrive before $t\in T$ and the number of vehicles is_s that were initially in-charge at charging station $s\in S$.

$$\sum_{k=0}^{t}\left(\sum_{i\in I|L_i\in Z}x_{is}a_{isk}-\sum_{i\in I|L_i=s}\sum_{d\in Z}x_{idk}\right)+is_s+ps_{st}\leqslant cp_s,\forall s\in S,t\in T \tag{45}$$

Constraints (46) and (47) ensure that each vehicle is assigned to at most one destination.

$$\sum_{d\in Z}x_{id}+\sum_{d\in S}x_{id}\leqslant 1,\ \forall i\in I|L_i\in Z \tag{46}$$

$$\sum_{d\in Z}\sum_{t\in T}x_{idt}\leqslant 1,\ \forall i\in I|L_i\in S \tag{47}$$

The objective of algorithm C is to minimize the routing cost plus penalties associated with the SoC of the chosen vehicles (e.g. sending vehicles with high battery to charging or with low battery to relocation) plus the penalties for deviating from the plan of algorithm B. The charge consumption associated with vehicle $i\in I$ going to destination $d\in S\cup Z$ is given by c_{id}. The penalties for slack flow going-to-charge of charging type $c\in C$ and for slack relocation flow are given by β^c_c and β^r, respectively. These penalties control how closely, or to what cost, the plan of algorithm B is followed. In addition, β^i_c is the penalty for the slack number of vehicles in-charge of charging type $c\in C$ per time-step and β'^i the penalty for the surplus total number of vehicles in-charge. The higher β'^i, the more we value having free vehicles over restoring battery of the system in the cases that we have excess energy consumption compared to the estimations.

Next, cost α^r_i is associated with the SoC of vehicle $i\in I$ that is relocated by algorithm C, which is negatively proportional to the metric of the SoC group (see Section 3.4.2). This penalty prioritises using higher charged vehicles, if necessary. Cost α^c_i associated with the SoC of vehicle $i\in I$ that is sent to a charging station by algorithm C is proportional to the SoC of the vehicle. This penalty prioritises sending lower charged vehicles to chargers. Cost α^f_{it} is associated with the SoC of vehicle $i\in I$ that has to stop charging at time $t\in T$ and is calculated by ranking all the possible ways we can cut charging of the vehicles in-charge. Algorithm 2 shows how to obtain α^f_{it}.

$$\begin{aligned}\min&\sum_{i\in I}\sum_{d\in Z}x_{id}\left(c_{id}+\alpha^r_i\right)+\sum_{i\in I}\sum_{d\in S}x_{id}\left(c_{id}+\alpha^c_i\right)+\sum_{i\in I}\sum_{d\in Z}\sum_{t\in T}x_{idt}\left(c_{id}+\alpha^f_{it}\right)\\&+\sum_{c\in C}\sum_{o\in Z}\sum_{d\in E_c}\beta^c_c s^c_{od}+\sum_{c\in C}\sum_{t\in T}\beta^i_c s^i_{tc}+\sum_{c\in C}\sum_{t\in T}\beta'^i s'^i_{tc}+\sum_{o\in Z}\sum_{d\in Z}\beta^r s^r_{od}\end{aligned} \tag{48}$$

Algorithm 2.

Ranking rule for stopping charge actions

a_{it} is the action of stopping the charging of vehicle $i\in I$ at $t\in T$
A is the set of all actions a_{it}
$R(a_{it})$ is the rank of action $a_{it}\in A$
L_{i0} is the initial SoC of vehicle $i\in I$
L_{it} is the SoC of vehicle $i\in I$ at time $t\in T$
A^f is the average SoC of vehicles that stop charging given by algorithm A
Function $U(L_{it})$ maps L_{it} to its difference from A^f, where the differences of below 15% charge are neglected.

$$U(L_{it})=\begin{cases}0, & \text{if}|L_{it}-A^f|\leqslant 0.15\\ |L_{it}-A^f| & \text{otherwise}\end{cases}$$

if $L_{i0}>L_{j0}$ **then**
  $R(a_{it})<R(a_{jt})$
if $U(L_{it})>U(L_{ik})$ **then**
  $R(a_{it})>R(a_{ik})$
$\alpha^f_{it}\propto R(a_{it})$

3.4. Choosing vehicle for a request (Algorithm R)

3.4.1. General scheme of algorithm R

The first and most important function of algorithm R is to avoid assigning a request to a vehicle if the vehicle cannot reach a charging station after servicing its passengers. Therefore, when algorithm R receives k potential vehicles for a request, it has to check for each vehicle if the current SoC is higher than the combined consumption of the new tour (the tour including the new request), and the trip from the last point of the tour to its nearest charging station plus a 5% battery buffer to be on the safe side. As it is computationally expensive to find the closest charger every time a request is made, the areas in the city from which vehicles can get to a charger within 5, 10, and 15 km can be pre-calculated. Therefore, when a request is made, there is only the need to check if the point of interest (last drop-off of the vehicles) falls within the area of 5, 10, or 15 km (maximum) distance to the charging station. Two dependencies between the assignment and charging problem can affect the modeling of algorithm R. For the first dependency, consider two successive intervals (each of 30 min) in which the number of requests in the first time-step is n and $2n$ in the second time-step, and assume that each vehicle serves only one request. Now suppose that at the beginning of the first time-step, there are n vehicles that can only serve one request due to low battery and n vehicles that can serve two requests consecutively. Then, it is beneficial to give incentives to the assignment algorithm to use the n vehicles with higher SoC (that can serve requests in both time-steps) in the first interval, rather than the n vehicles with lower SoC. If the operator would use the n lower charged vehicles in the first time-step, the vehicles would be out of charge in the second time-step, which means that there would not be enough vehicles to serve all $2n$ requests in the second time-step. Limiting the set of vehicles in the first time-step increases the immediate routing cost, but it could compensate by not rejecting customers in the second time-step.

The core idea is that using the high SoC vehicles 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 another one with 0%). As mentioned previously, this can allow more requests to be satisfied or provide more flexibility during the time that vehicles must go to charge. This will potentially reduce the cost of electricity under variable prices. Therefore, an indicator is developed (see Section 3.4.2) which indicates when the policy of choosing high SoC vehicles more frequently would be crucial for the profitability of the operation.

The second dependency between assigning vehicles to clients and charging operations accrues in cases where the vehicle would have to charge right after dropping off a passenger. In this case, it is preferred that the drop-off location is close to a charging station. Therefore, the insertion cost of the request to the current tour of a vehicle can be combined with the go to charger cost to achieve a low overall routing cost. This completes all SoC related considerations that algorithm R needs in order to choose between the different available vehicles for a request.

Algorithm 3 describes how algorithm R evaluates and ranks the k options returned from the *Dispatcher* (see Fig. 1) 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. The SoC metric used by algorithm R is introduced in Section 3.4.2.

Algorithm 3.

Algorithm R

---

driving_range is the 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 the effect of the metric.
Get k vehicles from the *Dispatcher*. For each vehicle $i \in \{1, \ldots, k\}$, the position of the pick up and drop-off of the current request within the vehicle's existing tour is given.
For each candidate vehicle i, get the last drop-off coordinates d_i and d_i' before and after inserting the request, respectively.
For each candidate vehicle i, get the existing tour T_i and SoC L_i.
For each candidate vehicle i, calculate the consumption cost c_i and c_i' of the tour before and after inserting the request, respectively.
for *buffer* in [5, 10, 15, 20] **do**
  if distance from d_i to nearest charging station (km)< *buffer* **then**
    $ct_i = 100 \times \dfrac{buffer}{driving\_range}$
    break
for *buffer* in [5, 10, 15, 20] **do**
  if distance from d_i' to nearest charging station (km) < *buffer* **then**
    $ct_i' = 100 \times \dfrac{buffer}{driving\_range}$
    break
if $L_i < c_i' + ct_i' + 5$ **then**
  Eliminate candidate vehicle i
if $L_i - c_i' < 15$ **then**
  Vehicle i must charge after last drop-off, so $g_i = 1$.
Calculate the generalized cost as $C_i = c_i' - c_i + g_i(ct_i' - ct_i) + w_{metric} \times metric(L_i)$.
Choose vehicle i with minimum generalized costs C_i.

Note that in this work, we do not offer or recommend any dispatching algorithm. For our simulations for the case study in Section 4.3, we use *PTV MaaS Dispatcher*, which has a greedy approach on accepting customers. While *PTV MaaS Dispatcher* can select vehicles, we did not have access to its internal objective value. This internal objective value is approximated by algorithm R as the marginal energy consumption imposed on the vehicle. However, any operator using our Smart Charging approach can plug in their favorite dispatching algorithm and can add our charging algorithm on top of that. So, algorithm R obtains potential vehicles from the used dispatching algorithm and adds additional terms to the objective function of the dispatching algorithm or filters some vehicles which are not able to complete the trip. Note that in some markets a greedy approach might be inferior to the strategic holding of taxies (Yang et al., 2020).

Next to this, if the operator wants to refuse a customer based on other factors than the fleet's battery, e.g. the route is not likely to have more passengers to share the ride, it can just do so, and simply not call algorithm R. Also after acquiring the updated objective values from algorithm R, it is still possible to decide not to serve a customer based on the obtained energy related information. One option that is not supported by algorithm R is to select a higher charged vehicle in the case that it is expected that more passengers will join the ride, since we do not have data available to simulate and test this concept. In general, in our methodology, the charging plan is kept independent of other policies such as pricing schemes (except for average income per hour information given to algorithm A) and other business operational parameters which are known to affect the performance of the systems.

3.4.2. Metric for choosing vehicles with high SoC

The metric used in algorithm R should help identifying when it is critical to use the high SoC vehicles more frequently than the low SoC vehicles. Therefore, a high metric value is expected if three conditions are met at the same time. The first condition is that the remaining charge of the fleet should not be significantly higher than the energy consumption that is expected over the rest of the day. The second condition is that there is a nonuniform SoC distribution across vehicles, since if there is no variation in the SoC of the fleet, the SoC will not affect the assignment decision. The third condition is that the expected demand is increasing in the near future. Eqs. (49)–(51) present the formulation with which we derive this indicator. The metric in (50) can be interpreted as the ratio of vehicles with SoC above x that should serve a trip to all vehicles with SoC above x. The numerator in Eq. (49) is the minimum number of trips that vehicles with SoC above x would have to be assigned to from time t to $t^{'}$. The denominator is the number of such vehicles at time t times $t^{'}-t$, which hypothetically is the number of trips that all those vehicles could do, given that they were continuously employed until $t^{'}$. The metric is normalized via Eq. (51) to yield the relative emphasis on using each SoC group x. The principle behind normalization is that if the ratio of vehicles with SoC above x that should serve a trip is the same as the ratio of the vehicles above x to all potential vehicles, then the normalized metric should be one, suggesting that there is no need to promote the choice of vehicles with SoC above x.

$$p_{xt'} = \frac{T_x - (T_0 + T_c - T_d)}{n \times d} \tag{49}$$

$$p_x = max_{t'}(p_{xt'}) \tag{50}$$

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 that all current vehicles with SoC above x can serve until time $t^{'}$
T_0= Trips that all current vehicles can serve until $t^{'}$
T_c= Trips that all vehicles that come out of charge can serve until $t^{'}$
T_d= Trips that the operator wants to serve until $t^{'}$
n = Number of vehicles with SoC above x
d= $t^{'}-t$

$$P_x = max\left(\frac{\frac{p_x}{p_0}}{\frac{n_x}{n_0}} = \frac{p_x n_x}{p_0 n_0}, 1\right) \tag{51}$$

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)
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

4. Numerical simulation for the case study city of Barcelona

4.1. Case study design

The *Daily charging planner* and *Online charging planner*, consisting of algorithms *A*, *B*, and *C*, were implemented in Python and solved with commercial solver Gurobi Optimization (2019). The dispatcher used for the case study was *PTV MaaS Dispatcher* from PTV Group (Planning and Operating Mobility). *PTV MaaS 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 charge, and to make sure they do not run out of charge during operation. *PTV xroute* routes the vehicles between known waypoints (pickup, drop-off points, or any specified coordinate) and is used by the charging planner to get information about the distance of the vehicles to the chargers. Finally, *PTV MaaS Simulator* is used to mimic the real world, namely the road network and the passenger requests.

The city chosen as case study for this research is Barcelona. For trip generation, an hourly OD matrix was available which comprises 300 zones. Trip origin and destination coordinates were selected randomly from 50 potential locations in each of the zones. The

Table 2
Parameters case study.

Requests	Unit	Value
Operation start time	time of the day	6
Operation end time	time of the day	22
Expected No. requests	count	8500
Max waiting time	minutes	10
Chargers	**Unit**	**Value**
No. fast charging plugs	count	3
No. fast charging stations	count	2
No. slow charging plugs	count	20
No. slow charging stations	count	4
Fleet	**Unit**	**Value**
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

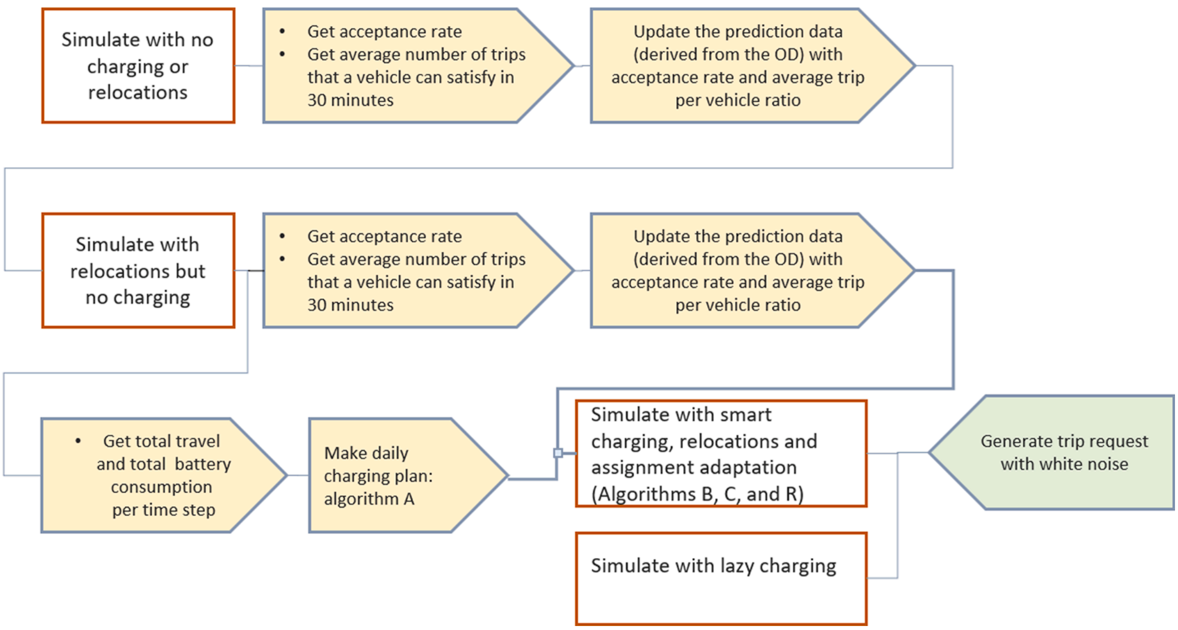


Fig. 5. Summary of simulation steps.

parameters used to configure the scenarios are summarized in Table 2. Zoning boundaries, as defined in algorithm *B*, were constructed by aggregating the 300 zones in the OD matrix. The fleet size and the size of the operation area dictate the zone size suitable for algorithm *B* to avoid many zones with few or no vehicles. In this case, for 150 vehicles, 14 zones were used. The historical operation data was emulated with executing two proxy simulations before the main simulation (Fig. 5). The trip requests for the proxy simulations were drawn randomly from the same OD matrix. Note that the OD matrix is only used by the simulator and it is not known to the algorithm.

The baseline scenario for the case study is taking a simplistic approach to charging, also referred to as Lazy Charging for the remainder of the paper. In the Lazy Charging algorithm, vehicles only move toward the charger if their SoC falls below 20% and recharge at the closest charging station up to a SoC of 90%. The vehicles will not drive to the second closest charger if the first choice is at capacity. The improved scenario is the Smart Charging approach described in the previous sections.

4.2. Performance

Experiments on the case study city were run on a computer with an Intel R processor with 8 cores running at 3600 MHz using 32 GB of RAM under Windows 10. The performance was tested for fleet sizes of 150, 1500, and 15,000, while the number of zones (used in algorithm *B*) was kept constant. The gap for algorithm *A* is unknown due to using the Column Generation heuristic (Puchinger et al., 2011). The gap tolerance of algorithm *B* is set to 10% and algorithm *C* is always optimal. Table 3 shows that the solving times of algorithms *A* and *B* remain independent of the fleet size. Algorithm *A* and *B* could be formulated by synthetically generated inputs, without running a full simulation, however, to measure the scaling of algorithm *C* actual simulations were required, which the hardware configuration did not allow. The number of decision variables grows linearly in algorithm *C*, but since the decisions are local and have a short horizon, their dependency remains constant. The solving times of algorithms *B* and *C* are interdependent and rely on the number of zones. For larger zone sizes, so smaller number of zones, algorithm *B* becomes less computationally costly at the expense of a longer preparation and solve time for algorithm *C*. In general, algorithm *B* has a better long term overview and algorithm *C* has richer information on the current state of vehicles, and therefore, changing the zone size would simultaneously change performance and solution quality.

4.3. Results

According to Table 4, the Smart Charging scenario had an 8% higher acceptance rate for requests than the Lazy Charging scenario, which means that using Smart Charging resulted in 14% more revenue for the operator. The higher acceptance rate is a joint result of relocating and having fewer rejections owing to not having enough charged vehicles. The number of rejected requests due to not

Table 3
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	–
15000	462.47	0.163	–

Table 4
Results.

	Unit	Lazy Charging	Smart Charging
Total No. requests	count	7740	7740
Accepted No. requests	count	4293	4905
Rejected for not having enough charge	count	702	238
Percentage accepted	percentage	55.4	63.37
Total distance driven	km	16199	20437
Total travel time of vehicles	hour	848	1061
Distance per request	km	3.77	4.17
Ratio of total travel time to sum of direct travel time	ratio	1.25	1.46
No. relocations	count	na	403
Average relocation distance	km	na	2.7
Total relocation distance	km	na	1104
Percentage of relocation trips that got a trip after relocating	percentage	na	78
No. trips to a charger	count	32	213
Average trip distance to charger and back	km	2.1	7
Total distance of trips to charger and back	km	67	1431
Electricity cost penalty	na	1388	4302
Total charge gained in slow charging	in ratio to one full battery	9.4	24.4
Total charge gained in fast charging	in ratio to one full battery	10	21.8

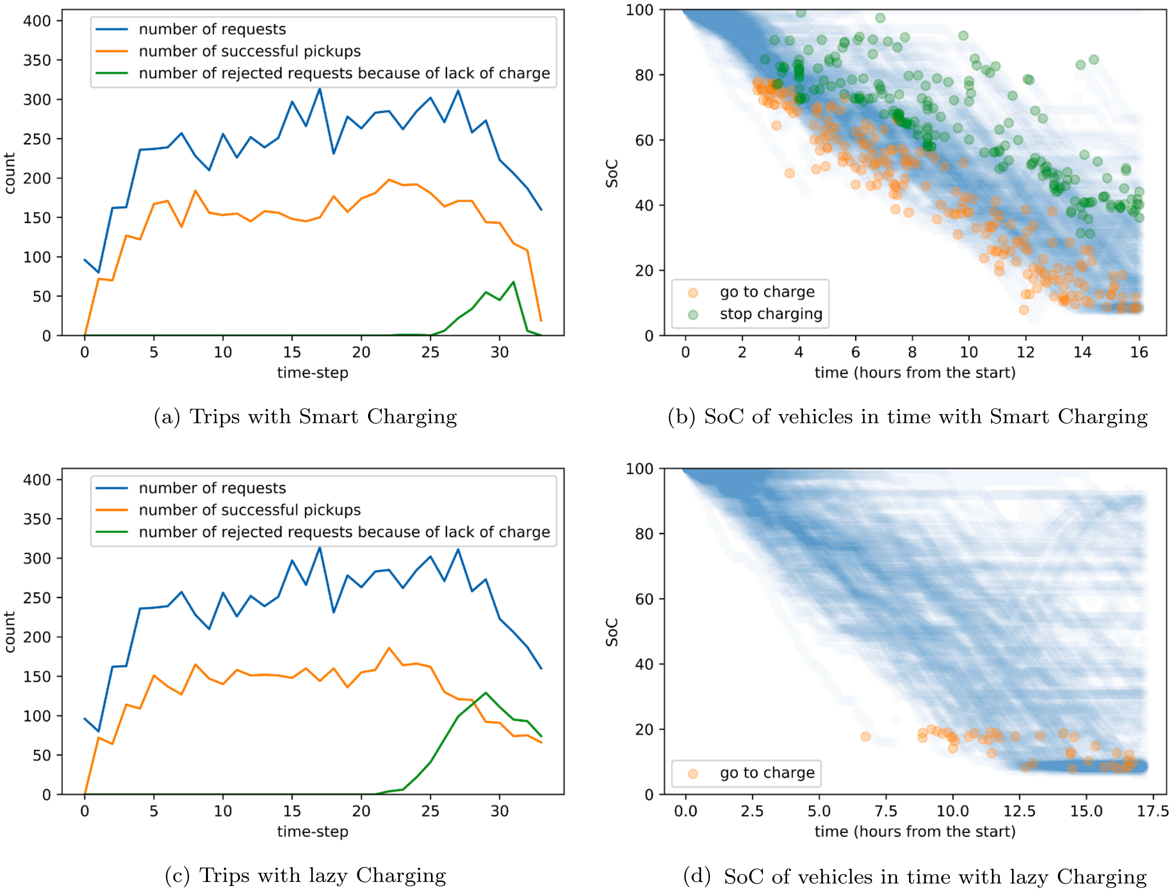
having enough charged vehicles decreased from 702 to 238 with Smart Charging and overall 612 more requests were satisfied. Distance per request and ratio of travel time to direct travel time are both measures of cost-effectiveness, and as expected, Lazy Charging is the best (lowest) in these measures, for which it compromises the acceptance rate.

The percentage of relocations which generated a client trip afterward is calculated by flagging the relocations as successful if the vehicles got a trip within 20 min from the time they were requested to relocate. The number of charging trips increases substantially with the use of Smart Charging. That is first because, with Smart Charging, the amount of recharged battery during the day was higher. To achieve this, Smart Charging has to use chargers earlier on in the day when vehicles' batteries 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 among the vehicles.

The average distance of going to the charger and back is around 7 km in all scenarios using Smart Charging and 2 km in the Lazy Charging scenario. The difference can be explained by two main reasons. First, the average distance to charger and back in the Smart Charging strategy is technically charging and a relocation trip, as after charging, a vehicle ends up where demand is expected. On the other hand, in the Lazy Charging, a 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, which increases the average cost, as the extra vehicles are likely to move from areas farther away from the chargers. The total distance to a charger and back is about 7% of the total distance traveled.

Fig. 6a shows the total number of customers who requested a ride in blue and the number of customers whose request was satisfied within the acceptable waiting time (10 min from making the request) in orange given the vehicle availability. The gap between the two lines then indicates the number of rejected customers, either due to vehicles not being available or not having enough battery. Therefore, a third green line has been added to count specifically rejections resulting from lack of battery. In the early hours, the 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 (corresponding to hour 13 in the operation) requests get rejected because of having not enough charged vehicles.

The sum of the number of rejected and successful requests should not necessarily be the same. Reminding that the green line represents the number of rejected requests because of lack of charge, there could still be rejections because there was no vehicle available that could satisfy the maximum waiting time.



(a) Trips with Smart Charging

(b) SoC of vehicles in time with Smart Charging

(c) Trips with lazy Charging

(d) SoC of vehicles in time with lazy Charging

Fig. 6. Fleet SoC and level of service.

Fig. 6b shows how the SoC of the fleet evolves during the day. The 150 vehicles start with a full battery, and their SoC decreases homogeneously, before the first charging trips. The SoC of vehicles going to charge is shown in orange and the SoC of vehicles coming out of charge is shown in green. The vertical distance between the green and orange line shows the average charge gained by vehicles in a charging trip, throughout the day.

Unlike Lazy Charging, Smart Charging does not have a hardcoded rule to send the lowest charged vehicles to charge. The preference to choose lower charged cars are coordinated through the objective function of algorithm *C*. In Fig. 6b, we can compare the SoC of all vehicles (blue) to the SoC of vehicles that are sent to charge (orange) and confirm that the lowest charged vehicles are sent to charge. This roughly verifies that we have chosen the right weight values in objective function of algorithm *C*.

Fig. 6c shows that, with Lazy Charging, the operator starts rejecting customers for not having enough charged vehicles from time-step 20, almost 2 h earlier than with Smart Charging. And finally, a significant gap emerges between the blue and orange lines towards the end of the day. In the last time-steps of Fig. 6d, we see that, compared to Smart Charging, many vehicles have batteries above 50%, either because they charged to 90% or they were not used from the start, while 75 vehicles with SoC lower than 15% accumulate in the bottom of the figure.

Smart Charging handles charging stations centrally. It can estimate the cost of routing in and out of them and has information on whether they are located where the demand is expected. When charging stations are unevenly distributed between the zones, it becomes more important to use Smart Charging rather than Lazy Charging. Furthermore, algorithm *B* is based on the assumption that the operator does not have prior knowledge on the distribution of the SoC of vehicles after the first time-step. This means that Smart Charging works well if there is a good circulation between the zones and if the location of charged vehicles is fairly independent of the location of chargers. In large operating areas where most trips occur within two or more mega zones instead of between them, the operator should solve the algorithm *B* separately for each mega zone.

In Fig. 7, it is observed that Smart Charging satisfies fewer customers than the scenario with infinite battery around time-step 15. This is due to a peak in the number of vehicles that were in-charge. Lazy Charging satisfies fewer customers, even when vehicles have enough charge because it does not relocate vehicles. By the end of the day, the gap between Smart Charging and Lazy Charging enlarges as most vehicles end up with no charge in the Lazy Charging scenario.

Figs. 8a, b, c, and d show the distribution of demand among the vehicles. Although most vehicles have 30 to 45 trips a day, some vehicles have less than ten trips, which is a result of the relocation algorithm not being suited to deal with stochasticity in demand. Fig. 8a shows that there were no pick ups in 35% of the total vehicle-hours. The maximum number of pickups in 30 min was 4 per vehicle. Fig. 8c shows that the maximum number of trips of a vehicle to a charger was 4, which could be irritating to the drivers, while 12% of all vehicles had no charging trip at all. Fig. 8d shows the number of relocation trips, which ranges mostly between 1 to 4 per vehicle, translating to 1 relocation trip for every 13 passenger trips.

The pink line in Fig. 9a–d shows the daily plan given by algorithm *A*. The figures compare the offline plan of algorithm *A* to the outcome of the numerical experiment, which means the execution of the daily plan by algorithms *B* and *C* based on real time information. There is a delay in vehicles reaching a charger, as both algorithms *A* and *B* assume that vehicles can immediately arrive at a charger. There is a lag coming out of charge because of the policy to penalize stopping charge outside of the planned range.

Fig. 9c reveals a gap between the plan of algorithm *A* and the actual number of vehicles going to the slow chargers. This is a consequence of the policy of keeping the vehicles in-charge until they reach a specified SoC, where the specified SoC is extracted from the plan of 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 that, as previous vehicles were kept in-charge due to the policy

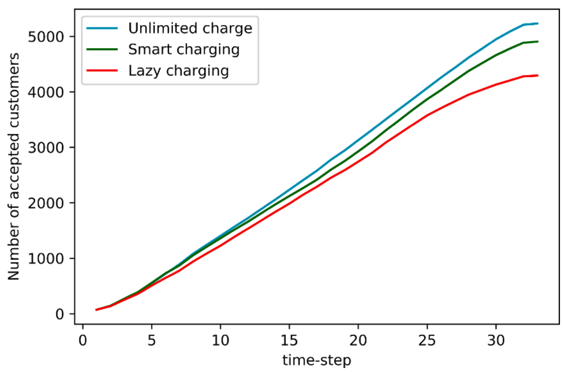
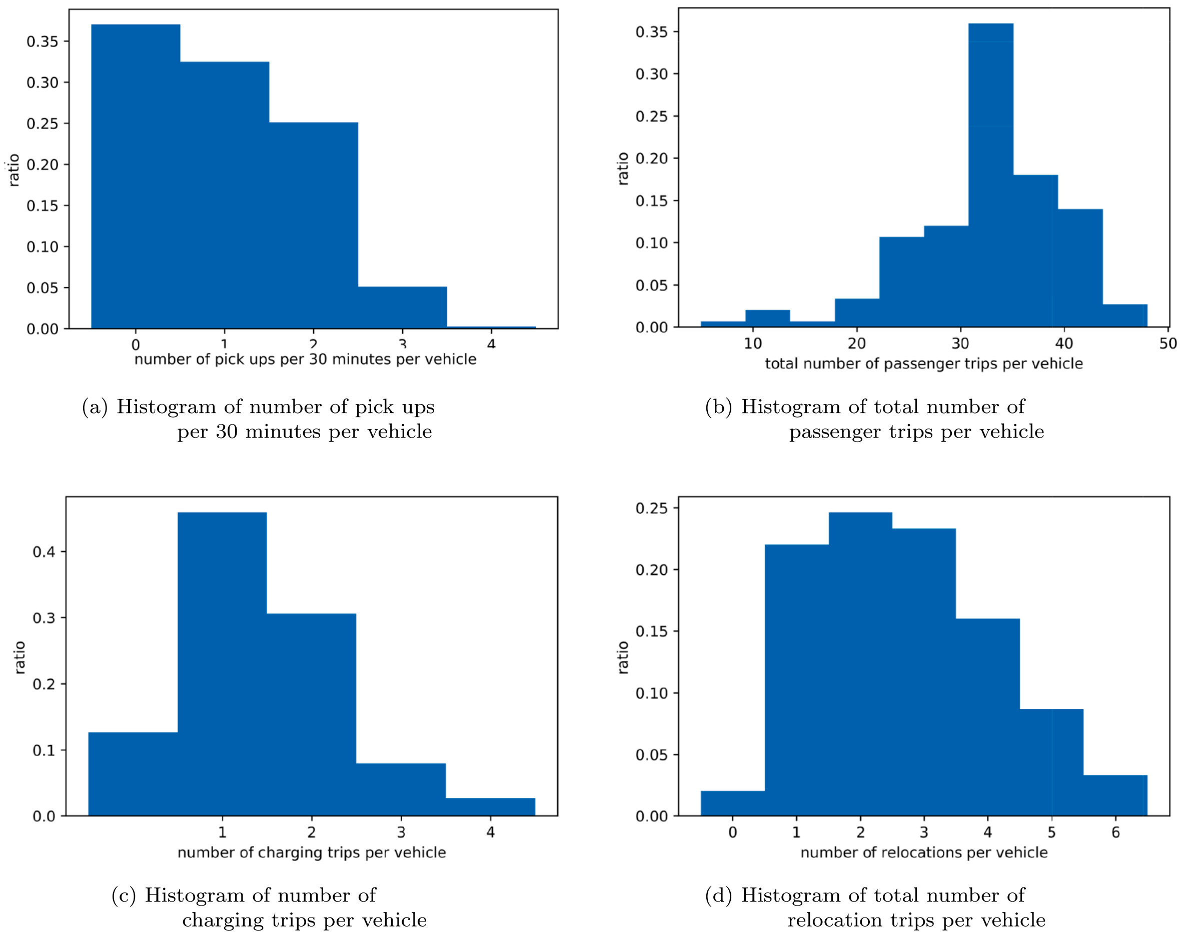


Fig. 7. Cumulative number of satisfied trips.



(a) Histogram of number of pick ups per 30 minutes per vehicle
(b) Histogram of total number of passenger trips per vehicle
(c) Histogram of number of charging trips per vehicle
(d) Histogram of total number of relocation trips per vehicle

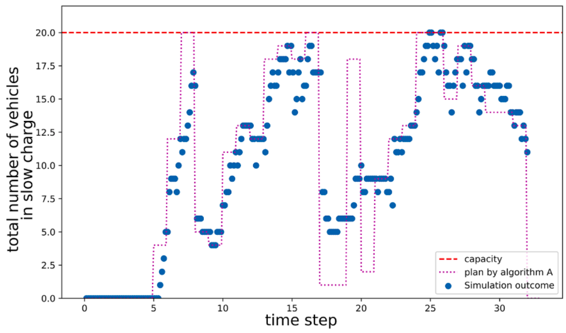
Fig. 8. Per vehicle statistics with Smart Charging.

discussed.

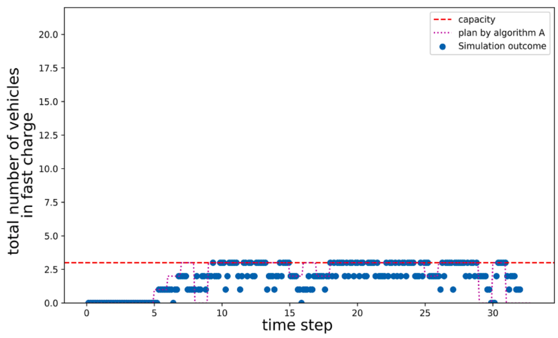
We tested the sensitivity of algorithm *B* and *C* to over and underestimation of demand. First, we increased the number of requests in the simulation by 30 percent without changing the daily planning, and zonal prediction of pickups and drop offs. In this scenario, the number of fulfilled trips increased by 8 percent while the charge gained increased by 8 percent. Second, we decreased the number of requests by 30 percent without changing the daily planning. In this scenario, the number of fulfilled trips was 17 percent lower while the charge gained by the fleet decreased by 13 percent. This shows algorithms B and C have some but limited flexibility to compensate for wrong daily plans, based on real time information. This is achieved by not only relying plans on the number of vehicles that should go to charge but also enforcing that the fleet should maintain a certain SoC. This is possibly more robust to fluctuations in demand rather than the exact number of vehicles in charge. The main disadvantage in the case with overestimation of demand was that the algorithm decreased the duration of charge rather than decreasing visits to the chargers. Dealing with demand stochasticity remains a challenging problem for future work in this area.

5. Conclusions and future work

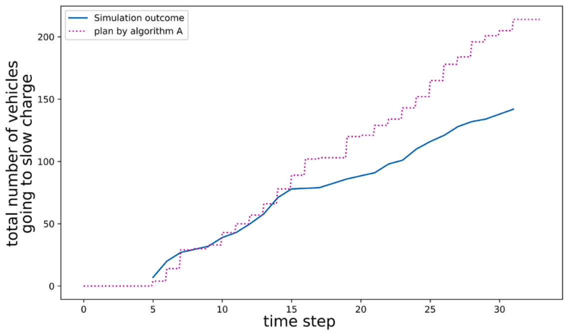
In this paper, an electric-charging planning algorithm was developed to accompany a dispatching algorithm that assigns real-time trip requests to vehicles belonging to a shared electric taxi fleet owned by an operator. The algorithm decides when, where, and how much each vehicle should charge, in real-time. The algorithm also relocates idle vehicles if needed. The design of the algorithm aimed to allow maximum flexibility for the 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), while providing enough charge for the expected level of demand to be met. Three sequential mixed integer linear programming (MILP) optimization models 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.



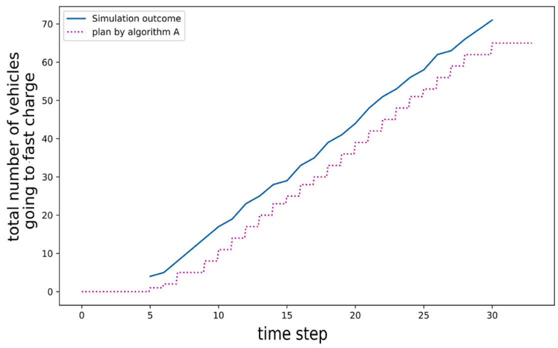
(a) Number of vehicles in slow charge



(b) Number of vehicles in fast charge



(c) Number of vehicles going to slow charge



(d) Number of vehicles going to fast charge

Fig. 9. Number of vehicles in charge and going to charge in simulation compared with daily plan by Algorithm A.

The first level (algorithm *A*) only decides on when to charge with time-steps of 30 min and a horizon until the end of the operational day. The decisions are re-optimized every four hours throughout the day. The second level (algorithm *B*) has time-steps of 30 min and a horizon of 2.5 h and is re-optimized every four minutes. It decides on an aggregated level where vehicles of each zone would charge or relocate to. The third level (algorithm *C*) is also re-optimized every four minutes and decides the actions of individual vehicles (e.g. going to charge, stopping to charge, and moving to another zone).

The computation time needed for the algorithm is within an acceptable range, namely less than 10 min for the daily planning and less than 10 s for the online planning. Employing column generation allows the performance of the daily planning to remain independent of the fleet size. The online planning, which is computationally more expensive due to the longer horizon, keeps a constant computation time independent of the fleet size.

Results show that the proposed charging algorithm can satisfy around 600 more trips (8% higher acceptance rate) than the Lazy Charging algorithm while spending far more travel time on the trips to the chargers. In the Smart Charging scenario, 7% of all distance traveled is for going to chargers and back, and 5% of all distance traveled is for relocation trips. Lazy Charging only invests 0.5% of the total distance traveled on trips to chargers. This means that, with Smart Charging, the operator added 14% to its revenue. By doing so, the average energy cost of a trip went up by 16%. To put the two percentages into perspective, we consider the revenue from riding a taxi for one kilometer in Barcelona at a minimum of 1.13 euros; while the electricity price for one kilometer is 0.05 euros. If originally the operator was spending 0.05 and gaining 1.13, with the improved algorithm, it will spend 0.066 and earn 1.29, showing a 13% improvement on the profit.

The main contribution of this study is the algorithm that offers the operator the ability to plan ahead, utilizing all the information available on future demand. The results of the case study show the potential of planning ahead, but do not analyze the cost and benefits of electric taxi operators. The higher driving distance per passenger is a cost that comes with servicing more clients. The trade-off between the average operation cost of the trips and the total number of satisfied customers can be managed by the operator, through changing the objective function weights in the daily charging planner (algorithm *A*). The potential improvement in the benefits will depend on electricity prices and taxi fare prices. The lower the electricity price and the higher the number of potential customers (per vehicle), the more improvement can be expected from employing Smart Charging. The designed charging algorithm can very well utilize charging stations close to 100% by replacing vehicles in time, without any queuing, unlike the Lazy Charging which could only use 75% of the slow chargers at the time when more than half of the fleet was out of charge.

The advantage of the developed algorithm is that the operator has the option to balance the number of requests that it aims to satisfy and the cost he is willing to pay for executing those requests. It also provides the operator with the ability to reschedule charging, and respond to known disruptions such as chargers being 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 provides options to the company, exploiting these options is not easy. Even with a simulator that can well replicate the real-world operation, determining the parameters that give the best outcome is burdensome. Besides, parameters are not a one fit all configuration. They will depend among other aspects on the fees and demand patterns.

There are several avenues for future research linked to the problem that was explored in this paper. For example, charging vehicles on a private network may not be realistic if the existing network is shared among different companies or with private drivers. Therefore, it is worth exploring how the taxi company's performance is affected by competiting for charging opportunities in a city. In addition, vehicles can have different specifications in terms of battery range or compatibility to the charging infrastructure which will depend on the specific case-study. The charging and discharging curves of a vehicle are non-linear by nature. The discharging depends on many factors such as slopes, vehicle mechanics, and speeds, whilst the charging depends on the charging station, temperature, and battery performance, to state a few. Therefore, being able to internalize this in an optimization model continues to be a challenge that needs to be addressed. Next to this, if electricity prices are too difficult to predict because the market becomes more dynamic with the growth of electric mobility, it may also be important to consider real-time electricity price variation throughout the day. Lastly, in this paper, we model real-time unknown requests based on statistical distributions that are not dependent on the real-time supply, that is, demand does not react to waiting times. Demand–supply interaction optimization models are much more complex since demand is by its nature non-linear making any optimization model harder to solve. Nevertheless, these models can provide a more realistic picture of the system performance, and are thus worth pursuing, possibly by simplifying other model components.

Credit authorship contribution statement

Helia Jamshidi: Conceptualization, Methodology, Software, Writing - original draft. **Gonçalo H.A. Correia:** Project administration, Methodology, Supervision, Writing - review & editing. **J. Theresia van Essen:** Supervision, Methodology, Writing - review & editing. **Klaus Nökel:** Supervision, Validation, Resources, Writing - review & editing, Funding acquisition.

Acknowledgement

The authors wish to express gratitude to N. Ahuja from PTV for sharing valuable insights. The authors thank the PTV Group for providing access to the *PTV Mass Modeller* and the *PTV xServer*, and for the financial support of this research.

Appendix A. Glossary

Algorithm A

Symbol	Description
I	The set of vehicles
T	The set of time-slots
C	The set of charging types
$t \in T$	Time-slot index
$i \in I$	Vehicle index
$c \in C$	Charging type index
D_t	The predicted number of required active vehicles at time $t \in T$
B_t	The predicted number of desired charged vehicles at time $t \in T$
L_{i0}	Initial charge of vehicle $i \in I$
L^1_{i0}	$\min(L_{i0}, 0.8)$
L^2_{i0}	$\max(L_{i0} - 0.8, 0)$
r^1_c	The charging rate of charging type $c \in C$ up to SoC of 80%
r^2_c	The charging rate of charging type $c \in C$ after SoC of 80%
e	The average energy consumption rate per time slot
M	The maximum number of visits to a charging station
N_c	The available number of chargers of type $c \in C$
s_{itc}	Binary variable that is 1 when vehicle $i \in I$ starts charging at a charger of type $c \in C$ at time $t \in T$
f_{itc}	Binary variable that is 1 when vehicle $i \in I$ stops charging at a charging station of type $c \in C$ at time $t \in T$
d_{it}	Binary variable that is 1 when vehicle $i \in I$ is serving demand at time $t \in T$
y_{itc}	Binary variable that is 1 when vehicle $i \in I$ is charging at a charger of type $c \in C$ at time $t \in T$
L^1_{it}	Real variable that takes the SoC of the battery of vehicle $i \in I$ at time $t \in T$ for the first fictitious battery
L^2_{it}	Real variable that takes the SoC of the battery of vehicle $i \in I$ at time $t \in T$ for the second fictitious battery
L^{1+}_{it}	Variable that takes the total charge of the first battery after charging during time $t \in T$ for vehicle $i \in I$ resulting from charge type $c \in C$
L^{2-}_{it}	Variable that takes the charge of the second battery of vehicle $i \in I$ after the battery consummation in time $t \in T$ has been deducted
u_{itc}	The excess charge of the first battery for vehicle $i \in I$ at time $t \in T$ resulting from charge type $c \in C$

(*continued on next page*)

(*continued*)

w_{it}	The excess charge of the second battery for vehicle $i \in I$ at time $t \in T$ (the charge over 100%)
a_{itc}	Binary variable that is 1 when the charge of the first battery of vehicle $i \in I$ at time $t \in T$ after charging at a charger of type $c \in C$ is more than 0.8
b_{it}	Binary variable that is 1 when L_{it}^{2-} is negative for vehicle $i \in I$ at time $t \in T$ and when vehicle $i \in I$ was serving demand
v_{it}	Variable that takes the amount by which the second battery is decharged in excess for vehicle $i \in I$ at time $t \in T$
q_{it}	Binary variable that is 1 if the SoC of vehicle $i \in I$ is higher than 20%, and it is not busy at time $t \in T$
Q_t	The slack variable for the number of required charged vehicles at time $t \in T$
β	The average revenue obtained from 30 min of a vehicle's operation
α_c	The average cost of going to a charging station of type $c \in C$
p_{tc}	The surcharge cost of charge gained with charging type $c \in C$ at time $t \in T$ in relation to the overnight charging price
γ	The penalty for the slack number of charged taxis
K	The set of daily plans that is given as input to the master problem
$k \in K$	Index for the daily plan
x_k	The number of vehicles that follow daily plan $k \in K$
C_k	The cost of daily plan $k \in K$
y_{tc}^{π}	The dual variable of vehicles in charging type $c \in C$ at time $t \in T$ in the master problem
d_t^{π}	The dual variable of active vehicles in the master problem at time $t \in T$
q_t^{π}	The dual variable of charged vehicles in the master problem at time $t \in T$

Algorithm B

Symbol	Description
T	The set of time-slots
C	The set of charging types
$t \in T$	Time-slot index
$c \in C$	Charging type index
Y_{tc}	The number of vehicles that have to be in-charge of type $c \in C$ at time $t \in T$
X_{tc}	The number of vehicles that have to start charging of type $c \in C$ at time $t \in T$
E_c	The set of charging zones for charging type $c \in C$
E	$\cup_{c \in C} E_c$
Z	The set of operation zones
v_{o1}	The number of vehicles present in zone $o \in Z \cup E$ at $t = 1$
x_{odt}	Non-negative integer variable that takes the number of vehicles relocating between operation zones $o \in Z$ and $d \in Z$ at time $t \in T$, going from operation zone $o \in Z$ to charging zone $d \in E$ at time $t \in T$, and moving back to operation zone $d \in Z$ from charging zone $o \in E$ at time $t \in T$
p_{ot}	The predicted number of pickups for each zone $o \in Z$ in time $t \in T$
d_{ot}	The predicted number of drop-offs for each zone $o \in Z$ at time $t \in T$
p_{ot}^{*}	Non-negative integer variable that takes the number of successful pickups from zone $o \in Z$ at time $t \in T$
s_{ot}	Non-negative slack integer variable for the number of pickups $o \in Z$ at time $t \in T$
v_{o1}	The initial number of vehicles in operation zone $o \in Z$
d_{ot}	The number of drop-offs predicted for zone $o \in Z$ at time $t \in T$
m_o	The charger capacity of charging zone $o \in E$
$B_{tc}^{'}$	Non-negative integer variable for the surplus of vehicles in-charge at $t \in T$ and charging type $c \in C$
B_{tc}	Non-negative integer variable for the slack variable for the number of vehicles in-charge for charging type $c \in C$ at time $t \in T$
G_{ct}	Non-negative integer variable for the slack number of vehicles that have to start charging with type $c \in C$ at time $t \in T$
L_t^{2+}	The upper bound SoC for being in-charge at time $t \in T$
L_t^{2-}	The upper bound SoC for vehicles that can be send to charge at time $t \in T$
v_o^{+}	The number of vehicles in zone $o \in Z$ that have enough charge for at least one trip at time $t = 1$
L_t^{-}	The lower bound SoC for vehicles that stop charging at time $t \in T$
v_{ot}^{-}	The number of vehicles in charging zone $o \in E$ that were put to charge before $t = 1$ and will reach L_t^{-} at time $t \in T$
MCD_c	The minimum charging duration for charging type $c \in C$
e_{od}	The average cost of going from one zone to the other for $o, d \in Z \cup E$
α	The penalty for an unsatisfied pickup
β_c	The penalty for the unsatisfied number of in-charge vehicles for charging type $c \in C$
θ_c	The penalty for the unsatisfied number of going to charge vehicles for charging type $c \in C$
$\beta^{'}$	The penalty for the number of surplus vehicles in-charge

Algorithm C

Symbol	Description
I	The set of vehicles
T	The set of time-slots
C	The set of charging types
$t \in T$	Time-slot index
$i \in I$	Vehicle index
$c \in C$	Charging type index
Z	The set of operation zones
E_c	The set of charging zones of charging type $c \in C$
E	$\cup_{c \in C} E_c$
$e \in E$	Set containing the charging stations within this zone
S_c	The set of charging stations of charging type $c \in C$
S	$\cup_{c \in C} S_c$
$s \in S$	Charging station is an element of exactly one charging zone $e \in E$
x_{id}	Binary variable that is equal to 1 when vehicle $i \in I$ is directed to operation zone $d \in Z$ or charger $d \in S$ after its last drop-off and 0 otherwise
x_{idt}	Binary variables that is equal to 1 when vehicle $i \in I$ that is in-charge has to move to operation zone $d \in Z$ at time $t \in T$ and 0 otherwise
B^c_{od}	The charging flow from operation zone $o \in Z$ to charging zone $d \in E$, from algorithm B
B^r_{od}	The relocation flow with $o, d \in Z$
B^f_{od}	The flows from charging zone $o \in E$ to operation zone $d \in Z$
B^i_c	The total number of vehicles planned to be in-charge of charging type $c \in C$
B^c_{od}	The number of vehicles that should go from operation zone $o \in Z$ to charging zone $d \in E$
L_i	The last drop-off location $L_i \in Z$ of vehicle $i \in I$ or the charging station $L_i \in S$ vehicle $i \in I$ is in
s^r_{od}	Non-negative slack variables for the relocation flows with origin $o \in Z$ and destination $d \in Z$
s^c_{od}	Non-negative slack variables for the charging flows with origin $o \in Z$ and destination $d \in S$
a_{ist}	Parameter which is 1 when vehicle $i \in I$ arrives at charging station $s \in S$ before time $t \in T$ and 0 otherwise
s^i_{tc}	Non-negative integer variable for the slack number of in-charge vehicles with charging type $c \in C$ at time $t \in T$
$s^{\prime i}_{tc}$	Non-negative integer variable for the surplus number of in-charge vehicles with charging type $c \in C$ at time $t \in T$
pc^i_{tc}	The number of planned trips to charging stations of charging type $c \in C$ that arrive before time $t \in T$ and the number of vehicles ic_c that were initially in-charge with charging type $c \in C$
cp_s	The capacity of charging station $s \in S$
ps_{st}	The number of planned trips to charging station $s \in S$ that arrive before time $t \in T$
is_s	The number of vehicles that were initially in-charge at charging station $s \in S$
c_{id}	The charge consumption associated with vehicle $i \in I$ going to destination $d \in S \cup Z$
β^c_c	The penalties for slack flow going-to-charge of charging type $c \in C$
β^r	The penalties for slack relocation flow
β^i_c	The penalty for the slack number of vehicles in-charge of charging type $c \in C$ per time-slot
$\beta^{\prime i}$	The penalty for the surplus total number of vehicles in-charge
α^r_i	The cost associated with the SoC of vehicle $i \in I$ that is relocated by algorithm C
α^c_i	The cost associated with the SoC of vehicle $i \in I$ that is sent to a charging station by algorithm C
α^f_{it}	The cost associated with the SoC of vehicle $i \in I$ that has to stop charging at time $t \in T$

References

Al-Kanj, L., Nascimento, J., Powell, W.B., 2020. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. Eur. J. Oper. Res.

Bellman, R., 1957. A Markovian decision process. Indiana Univ. Math. J.

Benedict Evans, 2017. cars-and-second-order-consequences.

Birrell, S.A., Wilson, D., Yang, C.P., Dhadyalla, G., Jennings, P., 2015. How driver behaviour and parking alignment affects inductive charging systems for electric vehicles. Transp. Res. Part C: Emerg. Technol. 58 (PD), 721–731.

Chen, Z., Liu, W., Yin, Y., 2017. Deployment of stationary and dynamic charging infrastructure for electric vehicles along traffic corridors. Transp. Res. Part C: Emerg. Technol. 77, 185–206.

Chen, T., Zhang, B., Pourbabak, H., Kavousi-Fard, A., Su, W., 2018. Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems. IEEE Trans. Smart Grid 9 (4), 3563–3572.

Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs. Oper. Res. 8 (1), 101–111.

Dong, J., Liu, C., Lin, Z., 2014. Charging infrastructure planning for promoting battery electric vehicles: An activity-based approach using multiday travel data. Transp. Res. Part C: Emerg. Technol. 38, 44–55.

Franco, J.F., Rider, M.J., Romero, R., 2014. An MILP model for the plug-in electric vehicle charging coordination problem in electrical distribution systems. In: 2014 IEEE PES General Meeting — Conference & Exposition. IEEE, pp. 1–5.

Gurobi Optimization, L., 2019. Gurobi Optimizer Reference Manual.

He, F., Yin, Y., Zhou, J., 2015. Deploying public charging stations for electric vehicles on urban road networks. Transp. Res. Part C: Emerg. Technol. 60, 227–240.

Iacobucci, R., McLellan, B., Tezuka, T., 2019. Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid. Transp. Res. Part C: Emerg. Technol. 100, 34–52.

Jung, J., Jayakrishnan, R., Choi, K., 2012. Shared-Taxi Operations with Electric Vehicles. Tech. rep.

Li, S., Huang, Y., Mason, S.J., 2016. A multi-period optimization model for the deployment of public electric vehicle charging stations on network. Transp. Res. Part C: Emerg. Technol. 65, 128–143.

Lu, J.-L., Yeh, M.-Y., Hsu, Y.-C., Yang, S.-N., Gan, C.-H., Chen, M.-S., 2012. Operating electric taxi fleets: A new dispatching strategy with charging plans. In: 2012 IEEE International Electric Vehicle Conference, IEEE, pp. 1–8.

Nicolas, L., Moura, S.J., 2016. Optimal Routing and Charging of Electric Ride-Pooling Vehicles in Urban Networks. PhD thesis, UC Berkeley: Energy, Controls, and Applications Lab.

Planning and Operating Mobility as a Service — PTV Group.

Puchinger, J., Stuckey, P.J., Wallace, M., Brand, S., 2011. Dantzig-Wolfe Decomposition and Branch-and-Price Solving in G12. Tech. rep.

Riemann, R., Wang, D.Z., Busch, F., 2015. Optimal location of wireless charging facilities for electric vehicles: flow capturing location model with stochastic user equilibrium. Transp. Res. Part C: Emerg. Technol. 58 (Part A), 1–12.

Sassi, O., Cherif-Khettaf, W.R., Oulamara, A., 2015. Multi-start Iterated Local Search for the Mixed Fleet Vehicle Routing Problem with Heterogenous Electric Vehicles. Springer, Cham, pp. 138–149.

Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48 (4), 500–520.

Tseng, C.-M., Chau, S.C.-K., Liu, X., 2018. Improving viability of electric taxis by taxi service strategy optimization: a big data study of New York city. IEEE Trans. Intell. Transp. Syst. 1–13.

Tu, W., Li, Q., Fang, Z., Shaw, S.l., Zhou, B., Chang, X., 2016. Optimizing the locations of electric taxi charging stations: a spatial–temporal demand coverage approach. Transp. Res. Part C: Emerg. Technol. 65, 172–189.

Wang, G., Xie, X., Zhang, F., Liu, Y., Zhang, D., 2018. bCharge: Data-driven real-time charging scheduling for large-scale electric bus fleets. In: 2018 IEEE Real-Time Systems Symposium (RTSS). IEEE, pp. 45–55.

Wang, G., Chen, X., Zhang, F., Wang, Y., Zhang, D., 2019. Experience: Understanding long-term evolving patterns of shared electric vehicle networks. In: The 25th Annual International Conference on Mobile Computing and Networking, vol. 25, pp. 1–12.

Yang, Z., Guo, T., You, P., Hou, Y., Qin, S.J., 2018. Distributed approach for temporal-spatial charging coordination of plug-in electric taxi fleet. IEEE Trans. Industr. Inf. 1.

Yang, H., Qin, X., Ke, J., Ye, J., 2020. Optimizing matching time interval and matching radius in on-demand ride-sourcing markets. Transp. Res. Part B: Methodol. 131, 84–105.

Zhu, M., Liu, X.Y., Wang, X., 2018. Joint transportation and charging scheduling in public vehicle systems – a game theoretic approach. IEEE Trans. Intell. Transp. Syst.