Schulz, Arne

Article

Using infeasible path cuts to solve Electric Vehicle Routing Problems with realistic charging functions exactly within a branch-and-cut framework

EURO Journal on Transportation and Logistics (EJTL)

Provided in Cooperation with:
Association of European Operational Research Societies (EURO), Fribourg

Suggested Citation: Schulz, Arne (2024) : Using infeasible path cuts to solve Electric Vehicle Routing Problems with realistic charging functions exactly within a branch-and-cut framework, EURO Journal on Transportation and Logistics (EJTL), ISSN 2192-4384, Elsevier, Amsterdam, Vol. 13, Iss. 1, pp. 1-14,
https://doi.org/10.1016/j.ejtl.2024.100131

This Version is available at:
https://hdl.handle.net/10419/325204

EURO Journal on Transportation and Logistics 13 (2024) 100131

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

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

Using infeasible path cuts to solve Electric Vehicle Routing Problems with realistic charging functions exactly within a branch-and-cut framework ☆

Arne Schulz

Institute of Operations Management, Universität Hamburg, Moorweidenstraße 18, 20148 Hamburg, Germany

ARTICLE INFO

Keywords:
Electric Vehicle Routing Problem
Mixed-integer programming
Branch-and-cut
Infeasible path cuts

ABSTRACT

The paper investigates the Electric Vehicle Routing Problem with a non-linear concave and strictly monotonic increasing charging function. In the literature, the non-linear charging function is typically approximated by a piecewise linear charging function which does not overestimate the real charging function in any point. As the piecewise linear charging function underestimates the real state-of-charge in some points, such an approximation excludes feasible solutions from the solution space. To overcome this drawback we introduce a new method to determine a piecewise linear charging function overestimating the real charging function in a way that the area between both functions is minimized as well as an adaptation of a known linearization to include the piecewise linear charging function in a branch-and-cut approach. Thereby, we include infeasible solutions in the solution space. To declare them infeasible again we check every integer solution obtained in the branch-and-cut procedure and add an infeasible path cut if the solution is infeasible for the real charging function such that the procedure terminates with an optimal solution for the real charging function. Our approach is evaluated in a computational study in which instances with up to 100 customers were solved to optimality. Moreover, we evaluate the trade-off between a more complex model formulation due to more binary variables if the number of supporting points for the piecewise linear approximation is increased and the higher approximation error if fewer supporting points are used.

1. Introduction

Due to the importance of electric vehicles for the required transformation to a zero-emission world, Electric Vehicle Routing Problems (EVRPs) have been extensively investigated in recent years (Qin et al., 2021). One of the most challenging aspects in modeling an EVRP is the charging function. While in the first years authors assumed a linear charging function (Schneider et al., 2014; Desaulniers et al., 2016), the state-of-charge (SOC) increases non-linearly but concavely in real world charging processes (Marra et al., 2012). However, while linear charging functions can be implemented straightforward in linear mixed-integer programming models, non-linear charging functions cannot be implemented directly. Therefore, most of the authors model non-linear charging functions in linear mixed-integer programming models for the EVRP by a piecewise linear approximation of the charging function (see e.g. Montoya et al. (2017)).

Indeed, a model with a piecewise linear approximation is not exact any more. Thus, the paper at hand introduces a method using infeasible path cuts to overcome this disadvantage of the piecewise linear charging function approximation leading to an exact solution approach. Thereby, we are in line with a future research direction proposed in the review by Kucukoglu et al. (2021) who state that non-linear charging operations should be incorporated. The proposed method can be used by researchers to solve EVRPs exactly within a branch-and-cut framework. To the best of our knowledge this is the first exact solution approach for EVRPs with time windows and a realistic non-linear and concave charging function.

Infeasible path cuts have already been used intensively in different contexts in the operations research literature. Primarily, they were introduced by Dantzig et al. (1954) to avoid subtours in the Traveling Salesman Problem (TSP). The basic idea is to identify a path within a tour of the current solution which is infeasible. Then, a cut is added ensuring that only $n-1$ of the n sequence variables determining the path can be set to 1 in a feasible solution. By this, the current infeasible solution is declared to be infeasible. In the paper, we use a piecewise linear approximation of the charging function which overestimates the real SOC. If the found solution in a branch-and-cut node is infeasible with respect to the real charging function, an infeasible path cut is added to declare the solution to be infeasible. By this, the best found solution is feasible with respect to the real charging function such that

☆ This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
E-mail address: arne.schulz@uni-hamburg.de.

https://doi.org/10.1016/j.ejtl.2024.100131
Received 30 June 2023; Received in revised form 25 January 2024; Accepted 18 March 2024
Available online 20 March 2024

we have found a feasible and optimal solution when the procedure terminates.

Summarized, our contribution is fourfold:

- We introduce a new method to determine a piecewise linear approximation which never underestimates the real charging function. As we want to minimize the number of obtained infeasible solutions in the solution process, we determine this piecewise linear charging function such that the area between the approximated and the real charging function is minimized.
- We present a mixed-integer program to solve the EVRP including an adaptation of the linearization of the charging function presented in Zuo et al. (2019) to include the overestimation of the real charging function.
- We use the model in a branch-and-cut framework. Thereby, we add infeasible path cuts whenever we obtain an integer solution which is infeasible regarding the real charging function such that the procedure terminates with an optimal solution regarding the real charging function.
- We evaluate the branch-and-cut algorithm in a computational study with up to 100 customers.

The paper is structured as follows: First, we review the literature regarding EVRPs with a focus on the charging process (Section 2). Then, we formally describe the EVRP (Section 3). In Section 4, we discuss the non-linear charging function before we introduce the solution approach in Section 5. Thereby, we introduce the used mixed-integer program (Section 5.1), including a linearization (Section 5.2) and preprocessing steps to improve the search (Section 5.3). Moreover, the separation procedure to identify infeasible paths (Section 5.4) is presented. Afterwards, our solution approach is evaluated in a computational study in Section 6. Finally, the paper closes with a conclusion (Section 7).

2. Literature review

The EVRP is a variant of the Vehicle Routing Problem (VRP) with electric vehicles which require charging. We refer to Pelletier et al. (2017) for an profound analysis of the special properties of electric vehicles in comparison to conventional vehicles. A general variant of the VRP for alternative fuel-powered vehicles is the green VRP (Erdoğan and Miller-Hooks, 2012). Our literature review focuses on the modeling of the charging process in linear mixed-integer programs. The charging process is typically either assumed to be linear or non-linear but concave. While a linear charging process can be modeled directly by constraints of the form

$$\text{SOC}_{\text{departure}_i} \leq \text{SOC}_{\text{arrival}_i} + \text{charging\_rate} \cdot (\text{time}_{\text{departure}_i} - \text{time}_{\text{arrival}_i})$$

with the linear charging rate for every location i, a non-linear concave function needs to be approximated to linearize it. This is typically done by a piecewise linear charging function underestimating the real SOC. For an example compare Fig. 1. By this, authors ensure that their solutions are feasible regarding the real charging function, but it might be that their approximation cuts off the optimal solution. We discuss the linearization in more detail including a formulation of the corresponding constraints in Section 5.2.

Both variants have their advantages and disadvantages. Linear charging can be modeled exactly but does not fully picture real-world charging processes. Piecewise-linear approximations are a more precise approximation of the real-world but lead to more difficult model formulations and are still not exact. We discuss the literature on both aspects in Sections 2.1 and 2.2.

For further reading on different aspects of the EVRP literature we refer to the various reviews by Pelletier et al. (2016), Erdelić and Carić (2019), Ghorbani et al. (2020), Zhou et al. (2020), Kucukoglu et al. (2021), Qin et al. (2021), Xiao et al. (2021), and Ye et al. (2022) published in recent years.

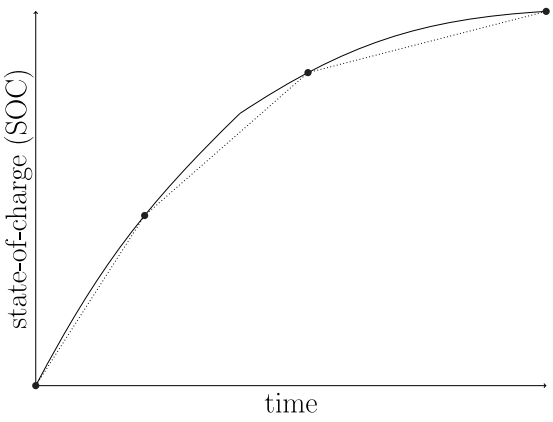


Fig. 1. Piecewise linear approximation of the real charging function with four supporting points (underestimation; Montoya et al. (2017)).

For conventional fossil fuel vehicles authors usually assume that the refueling process is fast enough such that the required time is neglected in solution approaches (see e.g. Schulz and Suzuki (2023)). Accordingly, under the assumption that refueling is fast and filling stations are widely available, standard formulations for the TSP (Dantzig et al., 1954; Miller et al., 1960) or the VRP (Toth and Vigo, 2014) even do not include any refueling process. In contrast, the charging process of electric vehicles requires a significant time such that it needs to be part of an EVRP model. In the following, we first review papers which assume charging to be linear (Section 2.1) and then those with non-linear charging (Section 2.2).

2.1. Linear charging

Early approaches for the EVRP considered linear charging functions (Schneider et al., 2014; Desaulniers et al., 2016), but also more recent approaches still model the charging process linearly. They typically make the simplifying assumption of linear charging to evaluate different aspects of routing problems with electric vehicles, which are not directly related to the charging process itself such that a linear approximation of the charging process is sufficiently precise. However, realistic non-linear charging could also be included in these approaches.

The focus in Hiermann et al. (2019) is on the mix of conventional, plug-in hybrid, and electric vehicles. Macrina et al. (2019) also considered a setting with a mix of conventional and electric vehicles. Moreover, their conventional vehicles had to follow a limit on emissions. In the work by Dönmez et al. (2022), multiple chargers are evaluated. The above mentioned charging time is used in Cortés-Murcia et al. (2019) to serve satellite customers with a different mode of transportation while the electric vehicle is bounded to the charging station. To avoid unused time while charging, charging time can also be reduced by reducing the required energy for a trip. Scholl et al. (2023) optimized platoon formations for long-haul transportation to reduce the required energy. Another option to avoid unused time during charging operations is to charge-while-driving. In the paper by Schwerdfeger et al. (2022), the electrification of roads with charge-while-driving technology is optimized. Boysen et al. (2023) then schedule point-to-point deliveries with overhead wiring. There is also an option to avoid unused charging time by swapping batteries. Verma (2018) and Raeesi and Zografos (2022) considered a problem variant where electric vehicles can charge their battery or swap the battery. Su et al. (2023) developed a deterministic annealing local search algorithm for the electric autonomous dial-a-ride problem.

From a methodical point of view several papers use Branch-and-Price algorithms with linear charging functions (Duman et al., 2022).

In the paper by Ceselli et al. (2021), a Branch-and-Price algorithm is developed for the EVRP with multiple technologies. A special aspect of their approach is that a column in the master problem represents a sequence of customer visits between two charging stations which needs not to be an entire vehicle's tour. The approach by Zang et al. (2022) further includes non-linear battery depreciation. The focus of Wu and Zhang (2021) is on a two-echelon setting.

2.2. Non-linear charging

If non-linear charging is included in EVRPs, authors typically approximate the charging function by a piecewise linear function (e.g. Kullman et al. (2021)). They thereby underestimate the SOC in dependency of the time. As a result the procedure terminates with a feasible solution but might also cutoff feasible solutions. Therefore, the solution is not necessarily optimal. The idea was introduced by Zündorf (2014) and included into a mixed-integer programming formulation by Montoya et al. (2017). The only study we found that also adds a further piecewise linear approximation of the charging function to overestimate the original charging function is Zhou et al. (2022). By this, the authors are able to evaluate the gap of their solution. In contrast, Montoya et al. (2017) indicated the average relative absolute error of their piecewise linear approximation to the original charging function. However, this does not allow to compute a gap, as the maximal error might be larger.

Further mixed-integer programming formulations for the EVRP with piecewise linear approximation of the charging function have been proposed by Froger et al. (2019) and Zuo et al. (2019). Kullman et al. (2021) and Froger et al. (2019) used the charging function approximation by Montoya et al. (2017). In Zuo et al. (2019), supporting points were equally distributed.

Camponogara and Nazari (2015) did not consider a non-linear charging function but a set of points of the charging function which are given. They presented several methods, including one which does not underestimate any charging point, to approximate them by a piecewise linear charging function minimizing the squared error. In our paper, we do not minimize the squared error or the maximal error between real charging function and approximation but introduce a new method to minimize the area included between real charging function and approximation. By this, we want to minimize the chance to obtain a solution which is infeasible due to the real charging function and, hence, minimize the number of required infeasible path cuts.

If we consider, from a methodical point of view, again Branch-and-Price algorithms, we have to differentiate between problem variants with and without time windows. Several authors used piecewise linear charging functions in problem variants with time windows (Lera-Romero et al., 2024; Liang et al., 2021; Klein and Schiffer, 2022; Lam et al., 2022). In problem variants without time windows, Branch-and-Price algorithms without the approximation of the non-linear charging function were presented (Xu and Meng, 2019; Lee, 2021; Zhang et al., 2021b). Without time windows no approximation is necessary, as due to the concavity of the charging function it is always superior to charge as late as possible. Thus, it is optimal to charge exactly enough to reach the next charging station. With time windows this is not possible, as the time window might be too short such that charging has to be pushed up to an earlier station. However, we do not know how much charging time is necessary at each charging station before the entire tour is determined. This makes the problem variant with time windows more challenging. Therefore, we include time windows in this paper.

As in the linear case most of the other studies using piecewise linear charging functions focus on different problem expansions. The approach by Pelletier et al. (2018) includes amongst others time-dependent energy costs and battery degradation. Battery degradation is also included in Guo et al. (2022), the energy consumption in Xiao et al. (2021). Load-dependent discharging is considered in Kancharla and Ramadurai (2020). Xu et al. (2022) focused on simultaneous pickups and deliveries. Capacity restrictions at the charging stations are considered in Froger et al. (2022) while Keskin et al. (2019) focused on time-dependent waiting times at charging stations and Koç et al. (2019) on the shared usage of charging stations among several companies. In the work by Zhang et al. (2021a), bus scheduling with a piecewise linear charging function is considered.

The literature review shows that there is a vast body of literature using linear or piecewise linear approximations of the charging function while we could not found any exact approach using a mixed-integer program to optimize an EVRP with time windows and non-linear charging. We close this research gap in the paper at hand.

3. Problem description

The EVRP extends the classical VRP. Thus, we consider a set $I = \{0, \dots, n+1\}$ of locations to be given. While locations 0 and $n+1$ represent the depot, locations $1, \dots, n$ are customer locations. Between each pair of locations $i, j \in I$ the travel time t_{ij} is known. We assume that the travel distance can directly be transformed into the travel time and that the travel time is deterministic. Moreover, we abstain from introducing a service time, as the service time at location i can simply be added to the travel times t_{ij} for all locations $j \neq i$. All locations $1, \dots, n$ have to be served by a number of K homogeneous vehicles which start and end their tours at the depot. Therefore, we define the binary variable x_{ij}, $i, j \in I$, which is 1 if a vehicle drives directly from location i to location j and 0 otherwise.

As we consider electric vehicles, the vehicles have a given battery capacity which we measure relatively. This means, the SOC of a battery is 1 if it is full and 0 if it is empty. Thus, we need parameters q_{ij} which indicate how much energy is required to drive between locations i and j, $i, j \in I$, in relation to the maximal battery capacity, i.e. $q_{ij} = t_{ij}/Q$, if the vehicle is able to drive for Q time units with a full battery. Note that Q can be smaller than the true physically possible driving range of the vehicles. By this, we can ensure that the battery is only used in a certain range to avoid battery degradation (Pelletier et al., 2017). Of course a battery is never allowed to exceed a SOC of 1 or fall below a SOC of 0. We model the SOC at locations $i \in I$ by the continuous variable s_i and assume that the battery is full when leaving the starting depot, i.e. $s_0 = 1$. Each vehicle can visit one of m charging stations $c \in C = \{1, \dots, m\}$ between any pair of locations $i, j \in I$. Like in Zuo et al. (2019) we assume for simplicity that a vehicle never charges directly after the starting depot. The assumption can be dropped by duplicating the depot once for every tour. The travel time between customer location i and charging station c is denoted as t_{ic} and t_{ci} for the opposite direction. The required energy is correspondingly denoted by $q_{ic} = t_{ic}/Q$ and $q_{ci} = t_{ci}/Q$, respectively. We assume the triangle inequality for the detours to charging stations to hold, i.e. that $t_{ic} + t_{cj} \geq t_{ij}$ and $q_{ic} + q_{cj} \geq q_{ij}$ holds for all pairs of locations i and j and all charging stations c. We do not require this, but it is possible that charging stations are at the same location as the depot or customer locations. This can simply be accomplished by setting the corresponding travel times to 0. We assume that only one charging station is visited between any pair of locations $i, j \in I$. At a charging station the vehicle charges its battery according to a non-linear charging function $f(\hat{t})$ depending on the time $\hat{t}$. We discuss the charging function in detail in Section 4.1. The binary variable y_{ic}, $i \in I$, $c \in C$, is 1 if a vehicle visits charging station c directly after location i and 0 otherwise. Our objective is to minimize the entire time spent on the vehicles' trips including travel and charging time.

As mentioned in Section 2 we focus on the problem variant with time windows. Thus, we assume a time window $[e_i, l_i]$ to be given for each location $i \in I \backslash \{0, n+1\}$ in which a vehicle has to start serving the customer. We abstain from restricting the maximal tour duration of the vehicles as it is done e.g. in Montoya et al. (2017). However, this can easily be included into the model by a further set of constraints.

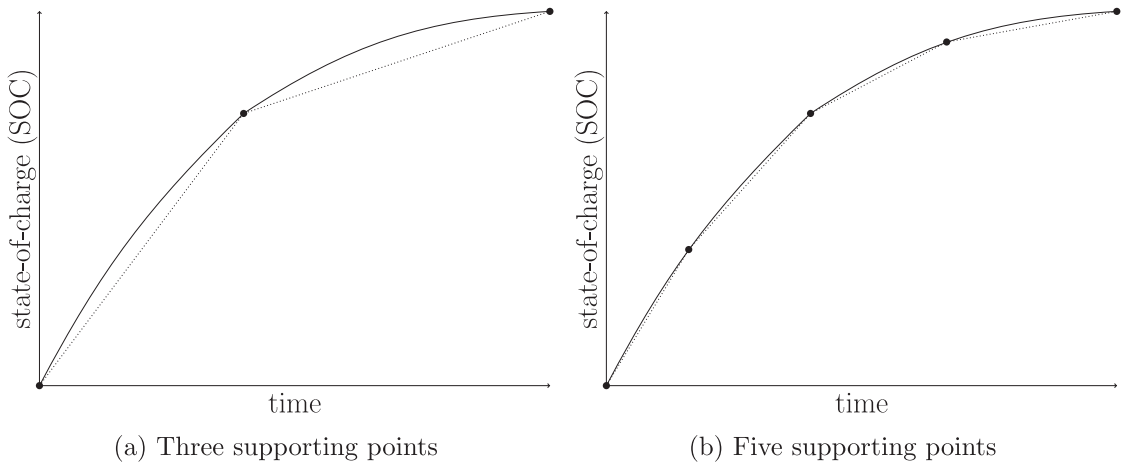


Fig. 2. Piecewise linear approximation of the charging function (underestimation; Montoya et al. (2017)).

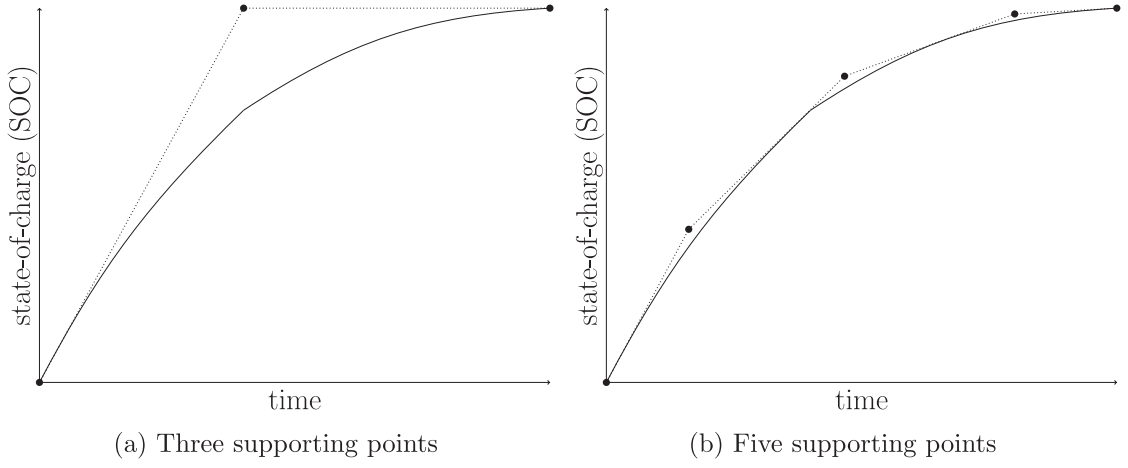


Fig. 3. Piecewise linear approximation of the charging function (overestimation).

4. Approximation of the charging function

In this section, we analyze the non-linear concave and strictly monotonic increasing charging function.

4.1. Non-linear concave and strictly monotonic increasing charging function

The charging process follows a non-linear concave and strictly monotonic increasing charging function in reality (Marra et al., 2012; Uhrig et al., 2015). We do not discuss the details on the charging process in this paper but refer to Marra et al. (2012), Uhrig et al. (2015), and Pelletier et al. (2017) for details. In this section, we focus on the modeling of the non-linear concave and strictly monotonic increasing charging function.

As stated in the literature review (Section 2) researchers typically approximate the charging function with a piecewise linear function whereat they underestimate the real charging function. Two examples are shown in Fig. 2. On the left side of the figure a piecewise linear approximation with three supporting points indicated by black circles – one at point (0,0), one at the point where the SOC is 1, and one somewhere in the middle – is shown (Fig. 2(a)). On the right side the same charging function is approximated with five supporting points (Fig. 2(b)). It can easily be seen that the approximation error, i.e. the area between charging function and approximation, becomes smaller with a larger number of supporting points (at least if they are chosen usefully). We will discuss on the position of the supporting points later in this section.

Fig. 2 shows a piecewise linear approximation where the original charging function is never overestimated. Analogously, Fig. 3 shows corresponding approximations such that the original charging function is never underestimated. This means that in Fig. 2 a found solution is always feasible, but we might ignore feasible solutions. In contrast, we might find an infeasible solution in Fig. 3 but will never declare a feasible solution to be infeasible.

One can easily think about an example where the disadvantage of both variants becomes clear. Let an instance with two locations and one charging station be given. Energy consumption and time windows can be selected such that the combination of charging time and required charge at the charging station, visited between the two locations, is between the piecewise linear function and the original charging function. Thus, the solution is either infeasible but declared as feasible (in case of overestimation) or feasible but declared to be infeasible (in case of underestimation).

Our idea is to overestimate the charging function (like in Fig. 3) such that we never declare a feasible solution to be infeasible. However, we might find an integer solution which is feasible for the charging function approximation but not for the original charging function. In this case, we add a cut to declare it as infeasible.

To minimize the chance to obtain an infeasible integer solution in the search process, we have to select the supporting points such that the area between the piecewise linear approximation and the original charging function is minimized while the piecewise linear approximation never underestimates the original charging function. As can be seen in Fig. 3 the approximation becomes the better the more

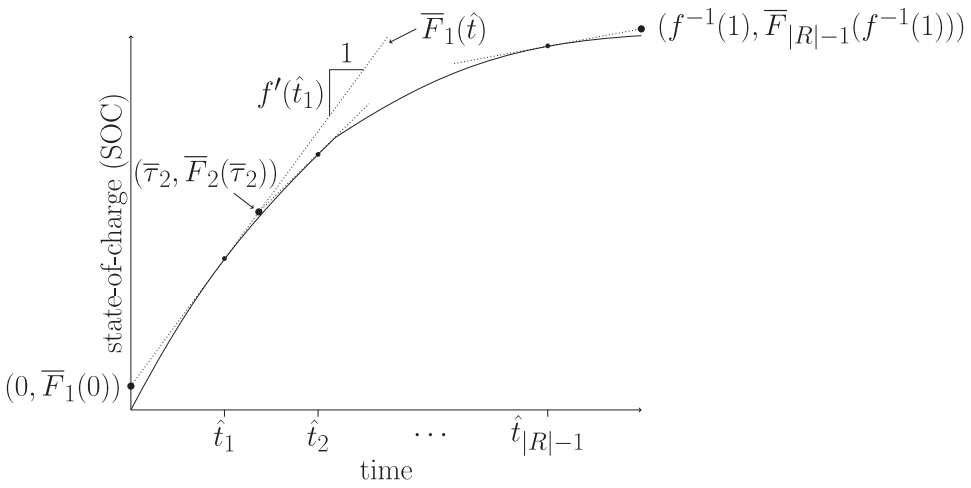


Fig. 4. Computation of starting solution.

supporting points are used. However, we will see in Section 5.2 that every added supporting point leads to additional binary variables in the model formulation. Hence, we have a trade-off between increasing the approximation error in favor of the model complexity and increasing the model complexity while decreasing the approximation error. We will investigate this trade-off in the computational study (Section 6) and assume for now the number of supporting points to be given when minimizing the approximation error.

In this paper, we assume the non-linear charging function to be given as a continuous concave and strictly monotonic increasing function like the one presented in Motoaki et al. (2018). However, in practice the charging function might not be given as closed form function but only at certain measuring points. For this case Camponogara and Nazari (2015) presented a dynamic program to determine a piecewise linear upper envelope with minimal squared error.

4.2. The approximation

The approximation consists of two parts. First, we determine a first approximation (starting solution) which is improved afterwards by reoptimizing the positions of two consecutive supporting points iteratively.

4.2.1. Determination of a first approximation

Let the charging function $f(\hat{t})$ as well as the number of supporting points $|R|$ and the considered time interval $[0, f^{-1}(1)]$ be given. Note that the inverse function f^{-1} of f exists, since f is strictly monotonic increasing.

We start with a starting approximation as illustrated in Fig. 4. It is determined as follows:

(a) Divide the considered time interval $[0, f^{-1}(1)]$ into $|R|$ equally long intervals.
(b) Compute for every point in time $\hat{t}$ in the intersection of the intervals the unique linear function $\overline{F}_r(\hat{t}) = \overline{b}_r + \overline{\beta}_r \cdot \hat{t}$ given by $\hat{t}$, $f(\hat{t})$, and $\overline{\beta}_r = f'(\hat{t})$ whereat $f'(\cdot)$ is the derivative of $f(\cdot)$ (if the derivative does not exist, $\overline{\beta}_r$ can be set such that $\overline{F}_r$ does not intersect with f in any point beside $\hat{t}$).

These piecewise linear functions are an approximation of $f(\cdot)$ whereat the supporting points are given by the intersection of $\overline{F}_r(\hat{t})$ and $\overline{F}_{r+1}(\hat{t})$, $r = 1, \dots, |R| - 2$, as well as $(0, \overline{F}_1(0))$ and $(f^{-1}(1), \overline{F}_{|R|-1}(f^{-1}(1)))$.

4.2.2. Reoptimization of the positions of two consecutive supporting points

Afterwards, we optimize the starting solution to minimize the area between the piecewise linear functions and the non-linear charging function. Thus, we search for $|R|$ supporting points and their corresponding linear functions $\overline{F}_r(\hat{t})$ crossing supporting points r and $r+1$ such that the area between the resulting piecewise linear function $\overline{F}(\hat{t}) = \min_{r \in R \setminus \{|R|\}} \overline{F}_r(\hat{t})$ and $f(\hat{t})$ is minimized whereat $\overline{F}(\hat{t}) \geq f(\hat{t})$ holds for all $\hat{t} \in [0, f^{-1}(1)]$. Formally, our objective is

$$\int_0^{f^{-1}(1)} \overline{F}(\hat{t})d\hat{t} - \int_0^{f^{-1}(1)} f(\hat{t})d\hat{t}. \tag{1}$$

As $\int_0^{f^{-1}(1)} f(\hat{t})d\hat{t}$ is constant, the objective reduces to minimize

$$\int_0^{f^{-1}(1)} \overline{F}(\hat{t})d\hat{t} = \sum_{r=1}^{|R|-1} \frac{1}{2} \cdot (\overline{F}_{r+1}(\overline{\tau}_{r+1}) - \overline{F}_r(\overline{\tau}_r)) \cdot (\overline{\tau}_{r+1} - \overline{\tau}_r) + \overline{F}_r(\overline{\tau}_r) \cdot (\overline{\tau}_{r+1} - \overline{\tau}_r). \tag{2}$$

In our procedure, we change $\overline{\tau}_r$ and $\overline{F}_r(\overline{\tau}_r)$, $r \in R \setminus \{|R|\}$, iteratively to minimize the sum on the right side of (2) such that $\min_{r \in R \setminus \{|R|\}} \overline{F}_r(\hat{t}) \geq f(\hat{t})$ holds for all $\hat{t} \in [0, f^{-1}(1)]$.

The idea of the procedure is to delete two supporting points $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ and reoptimize their position while fixing the remaining piecewise linear functions. Fig. 5 shows the situation. In the figure, four supporting points are given. We reoptimize the position of supporting points $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ by optimizing the position of touch point p where the piecewise linear function crossing $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ touches the original charging function. As the touch point and the derivative $f'(p)$ of the original charging function determine the piecewise linear function, the positions of $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ are determined as intersections with the previous and next piecewise linear function, respectively. Thus, we have to vary p such that (2) is minimized. Given the fixed remaining piecewise linear charging functions, this is equivalent to minimize the sum of areas A and B in Fig. 5 between the piecewise linear functions and the original charging function.

Due to the concavity of f, $f'(p') \leq f'(p)$ for $p' > p$, i.e. the slope of $\overline{F}_r(\cdot)$ does not increase if p increases. At the same time $\overline{\tau}_{r+1} - p$ decreases such that area B decreases if p increases. On the other side, $p - \overline{\tau}_r$ increases if p increases. As the slope of $\overline{F}_r(\cdot)$ does not increase, area A increases if p increases. Since A increases and B decreases for increasing p, it follows that the sum of A and B is a convex function in dependency of p with a unique minimum.

Algorithm 1. *(Determination of optimal p)*

1: ***function*** SET_TOUCH_POINT($\overline{\tau}_r$, $\overline{\tau}_{r+1}$, $f(\cdot)$, $f'(\cdot)$, $\overline{F}_{r-1}(\cdot)$, $\overline{F}_{r+1}(\cdot)$, ε)

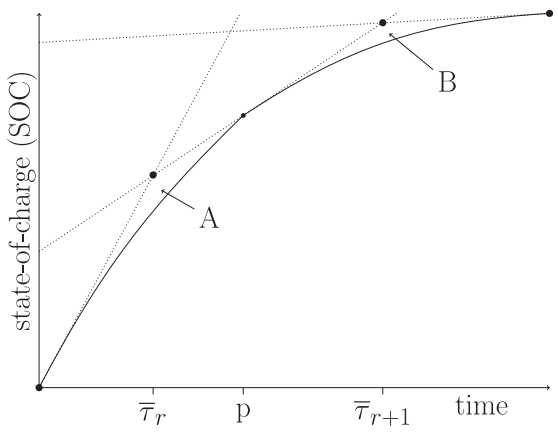


Fig. 5. Reoptimization of supporting points $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$.

2: *let* $\hat{p}$ *be the increasingly ordered list of possible touch points* $\hat{p} = \left(\overline{\tau}_r, \frac{\overline{\tau}_r+\overline{\tau}_{r+1}}{2}, \overline{\tau}_{r+1}\right)$
3: *set* $g(1) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\overline{F}_{r'+1}(\overline{\tau}_{r'+1}) - \overline{F}_{r'}(\overline{\tau}_{r'})) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'}) + \overline{F}_{r'}(\overline{\tau}_{r'}) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'})$ *whereat* $\overline{F}_r(\cdot)$ *does not change*
4: *set* $g(3) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\overline{F}_{r'+1}(\overline{\tau}_{r'+1}) - \overline{F}_{r'}(\overline{\tau}_{r'})) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'}) + \overline{F}_{r'}(\overline{\tau}_{r'}) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'})$ *whereat* $\overline{F}_r(\cdot) = \overline{F}_{r+1}(\cdot)$
5: *set* $g(2) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\overline{F}_{r'+1}(\overline{\tau}_{r'+1}) - \overline{F}_{r'}(\overline{\tau}_{r'})) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'}) + \overline{F}_{r'}(\overline{\tau}_{r'}) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'})$ *whereat* $\overline{F}_r(\cdot)$ *is the unique linear function determined by* $\frac{\overline{\tau}_r+\overline{\tau}_{r+1}}{2}$, $f\left(\frac{\overline{\tau}_r+\overline{\tau}_{r+1}}{2}\right)$, *and* $f'\left(\frac{\overline{\tau}_r+\overline{\tau}_{r+1}}{2}\right)$
6: *set current* $= 0$
7: *set best* $= \min\{g(1), g(2), g(3)\}$
8: ***while*** *best* $-$ *current* $> \varepsilon$ ***do***
9: *set best* $=$ *current*
10: $u = 1$
11: ***while*** $g(u) > g(u+1)$ *and* $u < |\hat{p}|$ ***do***
12: *set* $u = u + 1$
13: ***end while***
14: *insert* $\frac{\hat{p}_u+\hat{p}_{u+1}}{2}$ *in list* $\hat{p}$ *(*$\hat{p}_u$ *is the* $(u+1)^{th}$ *entry in extended list* $\hat{p}$*), evaluate the solution* $(\sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\overline{F}_{r'+1}(\overline{\tau}_{r'+1}) - \overline{F}_{r'}(\overline{\tau}_{r'})) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'}) + \overline{F}_{r'}(\overline{\tau}_{r'}) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'})$ *whereat* $F_r(\cdot)$ *is the unique linear function determined by* $\frac{\hat{p}_u+\hat{p}_{u+1}}{2}$, $f\left(\frac{\hat{p}_u+\hat{p}_{u+1}}{2}\right)$, *and* $f'\left(\frac{\hat{p}_u+\hat{p}_{u+1}}{2}\right)$ *and* $\overline{\tau}_r$ *and* $\overline{\tau}_{r+1}$ *are given as the intersection between* $\overline{F}_{r-1}(\cdot)$ *and* $\overline{F}_r(\cdot)$ *as well as* $\overline{F}_r(\cdot)$ *and* $\overline{F}_{r+1}(\cdot)$*), update* $g(\cdot)$*, and update current if improved*
15: *set* $u = |\hat{p}|$
16: ***while*** $g(u-1) < g(u)$ *and* $u > 0$ ***do***
17: *set* $u = u - 1$
18: ***end while***
19: *insert* $\frac{\hat{p}_{u-1}+\hat{p}_u}{2}$ *in list* $\hat{p}$ *(*$\hat{p}_u$ *is the* $(|\hat{p}| - u - 1)^{th}$ *entry in extended list* $\hat{p}$*), evaluate the solution* $(\sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\overline{F}_{r'+1}(\overline{\tau}_{r'+1}) - \overline{F}_{r'}(\overline{\tau}_{r'})) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'}) + \overline{F}_{r'}(\overline{\tau}_{r'}) \cdot (\overline{\tau}_{r'+1} - \overline{\tau}_{r'})$ *whereat* $\overline{F}_r(\cdot)$ *is the unique linear function determined by* $\frac{\hat{p}_{u-1}+\hat{p}_u}{2}$, $f\left(\frac{\hat{p}_{u-1}+\hat{p}_u}{2}\right)$, *and* $f'\left(\frac{\hat{p}_{u-1}+\hat{p}_u}{2}\right)$ *and* $\overline{\tau}_r$ *and* $\overline{\tau}_{r+1}$ *are given as the intersection between* $\overline{F}_{r-1}(\cdot)$ *and* $\overline{F}_r(\cdot)$ *as well as* $\overline{F}_r(\cdot)$ *and* $\overline{F}_{r+1}(\cdot)$*), update* $g(\cdot)$*, and update current if improved*
20: ***end while***
21: ***return*** $\overline{\tau}_r$, $F_r(\cdot)$, *and* $\overline{\tau}_{r+1}$ *determined by* $p = arg\ min_{p' \in \hat{p}}\{g(p')\}$.
22: ***end function***

Algorithm 1 uses this fact to find the optimal position of p and thereby supporting points $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ given the remaining supporting points $(\overline{\tau}_1, \overline{F}_r(\overline{\tau}_1)), \dots, (\overline{\tau}_{r-1}, \overline{F}_{r-1}(\overline{\tau}_{r-1}))$, $(\overline{\tau}_{r+2}, \overline{F}_{r+2}(\overline{\tau}_{r+2})), \dots, (\overline{\tau}_{|R|}, \overline{F}_{|R|-1}(\overline{\tau}_{|R|}))$ by nested intervals. The function starts with three potential positions for p (line 2) which are evaluated in lines 3–5. The best of them is our current best solution (line 7). In lines 8–20, we search for a new best solution by nested intervals until the improvement is smaller than a given ε. As we know that the solution space is convex and there is a single optimal solution, we run through the found potential positions for p stored in $\hat{p}$ once from the left side (lines 10–13) and once from the right side (lines 15–18). If there is a better solution, then it must be between this position u and the next one in the first case and u and the previous one in the second. The corresponding interval is bisected to determine a new potential position of p (lines 14 and 19, respectively). This new potential position of p is evaluated and added to $\hat{p}$ such that $\hat{p}$ is ordered increasingly. Function g, indicating the evaluation of $\hat{p}$ is updated accordingly. If the new p improves the current best solution, *current* is also updated. If we cannot find a solution which improves our current best one by more than ε, the algorithm returns $\overline{\tau}_r$, $\overline{F}_r(\cdot)$, and $\overline{\tau}_{r+1}$ such that the reoptimized supporting points $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ and $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ are set.

4.2.3. Approximation procedure

Algorithm 2 reoptimizes the supporting points iteratively with the help of Algorithm 1.

Algorithm 2. *(Determination of piecewise linear function* $\overline{F}(\hat{t})$*)*

1: *determine starting solution* $\rightarrow$ *current*
2: *set best* $=$ *current* $+ 1$
3: ***while*** *best* $-$ *current* $> \bar{\varepsilon}$ ***do***
4: *set best* $=$ *current*
5: ***for all*** $r = 1$ *to* $|R| - 1$ ***do***
6: *reoptimize supporting points* $(\overline{\tau}_r, \overline{F}_r(\overline{\tau}_r))$ *and* $(\overline{\tau}_{r+1}, \overline{F}_{r+1}(\overline{\tau}_{r+1}))$ *by* $SET\_TOUCH\_POINT(\overline{\tau}_r, \overline{\tau}_{r+1}, f(\cdot), f'(\cdot), \overline{F}_{r-1}(\cdot), \overline{F}_{r+1}(\cdot), \varepsilon)$
7: ***end for***
8: *evaluate current solution* $\rightarrow$ *current*
9: ***end while***

Algorithm 2 optimizes supporting points $(\overline{\tau}_1, \overline{F}_r(\overline{\tau}_1)), \dots, (\overline{\tau}_{|R|-1}, \overline{F}_{|R|-1}(\overline{\tau}_{|R|-1}))$, $(\overline{\tau}_{|R|}, \overline{F}_{|R|-1}(\overline{\tau}_{|R|}))$ and linear functions $\overline{F}_1, \dots, \overline{F}_{|R|-1}$ according to (2). Therefore, we first determine the starting solution as described above (line 1). Then, the solution is improved iteratively (lines 3–9) until the improvement is smaller than $\bar{\varepsilon}$. In each iteration, every pair of succeeding supporting points is updated in increasing order (lines 5–7) by Algorithm 1. Afterwards, the solution is evaluated and the current solution updated (line 8). Note that two supporting points are special. For the first we always set $\overline{\tau}_1 = 0$ and only update $\overline{F}_1(0)$. For the last we always set $\overline{\tau}_{|R|} = f^{-1}(1)$, i.e. to the time the battery is fully charged if the battery was empty when starting the charging process, and only update $\overline{F}_{|R|-1}(\overline{\tau}_{|R|})$. Moreover, Algorithm 2 requires function $\overline{F}_{|R|}(\cdot)$ in line 6 if $r = |R| - 1$. Let $\overline{F}_{|R|}(\cdot)$ be the unique linear function determined by $\overline{\tau}_{|R|}$, $f(\overline{\tau}_{|R|})$, and $f'(\overline{\tau}_{|R|})$ in this case.

5. Solution approach

In the following, we introduce our mixed-integer programming formulation in Section 5.1. Afterwards, we discuss the linearization of the model formulation (Section 5.2) and present some preprocessing techniques (Section 5.3). Finally, we introduce the separation procedure for our cuts (Section 5.4).

5.1. Mixed-integer program

Table 1 gives an overview of the used notation.

$$\min \sum_{i,j \in I} t_{ij} \cdot x_{ij} + \sum_{i \in I \setminus \{0,n+1\}} d_i + \sum_{i \in I} z_i \tag{3}$$

subject to

$$\sum_{j \in I} x_{ij} = 1 \quad \forall i \in I \setminus \{0, n+1\} \tag{4}$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in I \setminus \{0, n+1\} \tag{5}$$

Table 1
Notation.

Sets and indices	
$c \in C$	$= \{1, \dots, m\}$ charging stations
$h, i, j, k \in I$	$= \{0, \dots, n+1\}$ locations whereat 0 and $n+1$ represent the depot and $1, \dots, n$ are customer locations
$r \in R$	$= \{1, \dots, \lvert R \rvert\}$ supporting points of the piecewise linear charging function
Scalars	
K	Number of available vehicles
$M_{ij}, \bar{M}_{ij}, \tilde{M}_i$	Sufficiently large numbers
m	Number of charging stations
n	Number of customer locations
Q	Number of time units which can be driven with a full battery
Parameters	
$\bar{b}_r$	Intercept of piecewise linear function crossing supporting points $(\bar{\tau}_r, f(\bar{\tau}_r))$ and $(\bar{\tau}_{r+1}, f(\bar{\tau}_{r+1}))$
$\underline{b}_r$	Intercept of piecewise linear function crossing supporting points $(\underline{\tau}_r, f(\underline{\tau}_r))$ and $(\underline{\tau}_{r+1}, f(\underline{\tau}_{r+1}))$
$\bar{\beta}_r$	Slope of piecewise linear function crossing supporting points $(\bar{\tau}_r, f(\bar{\tau}_r))$ and $(\bar{\tau}_{r+1}, f(\bar{\tau}_{r+1}))$
$\underline{\beta}_r$	Slope of piecewise linear function crossing supporting points $(\underline{\tau}_r, f(\underline{\tau}_r))$ and $(\underline{\tau}_{r+1}, f(\underline{\tau}_{r+1}))$
e_i	Beginning of time window at customer location i
$\bar{F}_r(\hat{t})$	$= \bar{b}_r + \bar{\beta}_r \cdot \hat{t}$ linear function crossing supporting points $(\bar{\tau}_r, f(\bar{\tau}_r))$ and $(\bar{\tau}_{r+1}, f(\bar{\tau}_{r+1}))$ dependent on time $\hat{t}$ for all $r = 1, \dots, \lvert R \rvert - 1$
$\underline{F}_r(\hat{t})$	$= \underline{b}_r + \underline{\beta}_r \cdot \hat{t}$ linear function crossing supporting points $(\underline{\tau}_r, f(\underline{\tau}_r))$ and $(\underline{\tau}_{r+1}, f(\underline{\tau}_{r+1}))$ dependent on time $\hat{t}$ for all $r = 1, \dots, \lvert R \rvert - 1$
$\bar{F}(\hat{t})$	$= \min_{r \in R \setminus \{\lvert R \rvert\}} \bar{F}_r(\hat{t})$ piecewise linear charging function
$\underline{F}(\hat{t})$	$= \min_{r \in R \setminus \{\lvert R \rvert\}} \underline{F}_r(\hat{t})$ piecewise linear charging function
$f(\hat{t})$	Non-linear charging function dependent on time $\hat{t}$
$f^{-1}(\cdot)$	Inverse function of the non-linear charging function, i.e. the function indicating the time required to charge from SOC = 0 to SOC equal to the argument of the function
$f'(\hat{t})$	The derivative of $f(\hat{t})$
l_i	End of time window at customer location i
q_{ic}	Required energy to travel from location i to charging station c (q_{ci} correspondingly)
q_{ij}	Required energy to travel from location i to location j
t_{ic}	Travel time between location i and charging station c (t_{ci} correspondingly)
t_{ij}	Travel time between locations i and j
$\bar{t}_i, \underline{t}_i$	Charging time
$(\bar{\tau}_r, f(\bar{\tau}_r))$	Supporting point
$(\underline{\tau}_r, f(\underline{\tau}_r))$	Supporting point
Variables	
a_i	Starting time of service at location i
α_{ir}	Binary variable which is 1 if charging starts with a SOC between $f(\underline{\tau}_r)$ and $f(\underline{\tau}_{r+1})$ and 0 otherwise
d_i	Detour after location i to travel to a charging station and back to main route
ϕ_i	Amount of charge charged at a charging station directly after location i
s_i	SOC when reaching location i
x_{ij}	Binary variable which is 1 if a vehicle drives directly from location i to location j and 0 otherwise
y_{ic}	Binary variable which is 1 if a vehicle visits charging station c directly after location i and 0 otherwise
z_i	Non-negative variable indicating the charging time at a charging station directly after visiting location i

$$\sum_{j \in J} x_{0j} \leq K \tag{6}$$

$$\sum_{c \in C} y_{ic} \leq 1 \quad \forall i \in I \setminus \{0, n+1\} \tag{7}$$

$$d_i \geq (t_{ic} + t_{cj} - t_{ij}) \cdot (x_{ij} + y_{ic} - 1) \quad \forall i, j \in I, c \in C \tag{8}$$

$$s_j \leq s_i - q_{ij} + \left(1 - x_{ij} + \sum_{c \in C} y_{ic}\right) \cdot (1 + q_{ij}) \quad \forall i, j \in I \tag{9}$$

$$s_j \leq s_i - q_{ic} - q_{cj} + \phi_i + (2 - x_{ij} - y_{ic}) \cdot (1 + q_{ic} + q_{jc}) \quad \forall i, j \in I, c \in C \tag{10}$$

$$z_i \geq f^{-1}(s_i - q_{ic} + \phi_i) - f^{-1}(s_i - q_{ic}) - \tilde{M}_i \cdot (1 - y_{ic}) \quad \forall i \in I, c \in C \tag{11}$$

$$e_i \leq a_i \leq l_i \quad \forall i \in I \setminus \{0, n+1\} \tag{12}$$

$$a_j \geq a_i + t_{ij} - M_{ij} \cdot (1 - x_{ij}) \quad \forall i, j \in I \tag{13}$$

$$a_j \geq a_i + t_{ic} + t_{cj} + z_i - \bar{M}_{ijc} \cdot (2 - x_{ij} - y_{ic}) \quad \forall i, j \in I, c \in C \tag{14}$$

$$x_{ij}, y_{ic} \in \{0, 1\} \quad \forall i, j \in I, c \in C \tag{15}$$

$$s_i - q_{ic} \cdot y_{ic} \geq 0 \quad \forall i \in I, c \in C \tag{16}$$

$$\phi_i, z_i \geq 0 \quad \forall i \in I \tag{17}$$

Objective function (3) minimizes the entire time spent by all vehicles, i.e. travel times between locations, detours for charging, and charging times. Constraints (4)–(6) are the classical VRP constraints. Due to Constraints (4) and (5) every location is visited and left exactly once, respectively. Constraints (6) ensure that at most K vehicles are used.

The next block of constraints (7)–(11) are the charging constraints. Because of Constraints (7) at most one charging station is visited directly after location i. Constraints (8) indicate the detour if charging station c is visited between locations i and j, which is required for the objective function. If location j is visited directly after location i without any charging stop in between, Constraints (9) update the SOC. If a charging station is visited between locations i and j, the same is done by Constraints (10). The required charging time is pictured by Constraints (11). We will not consider f^{-1} in detail, as we do not need it in the linearized model (compare Section 5.2).

The block of constraints (12)–(14) includes the time windows. Arrival times have to be in the time window (Constraints (12)). They are determined by Constraints (13) if the vehicle drives directly from location i to location j. If charging station c is visited in between, Constraints (14) determine the correct arrival time.

Finally, the last block of constraints (15)–(17) sets the variable domains. Constraints (15) are the binary constraints for x and y variables. Constraints (16) ensure that a charging station is always reached with a non-negative SOC. Constraints (17) are the non-negativity constraints for the amount of charge charged at a charging station and the charging time. Note that a_i (by Constraints (12)) and d_i (by Constraints (8)) are already bounded by the constraints. Note further that we can fix x_{ii}, x_{i0}, $x_{n+1,i}$ as well as y_{0c} and $y_{n+1,c}$ to zero. By this, we can also exclude some constraints. However, due to readability we abstain from doing so in the presentation of the constraints, as they stay still valid.

Before we can use the model, we have to linearize Constraints (11) and set the Big M values M_{ij}, $\bar{M}_{ij}$, and $\tilde{M}_i$. This is done in the following two sections.

5.2. Linearization

Model formulation (3)–(17) is not linear due to the non-linear inverse of the charging function in Constraints (11). To linearize the constraints, we approximate the non-linear charging function f by a piecewise linear function as introduced in Section 4 (compare Fig. 3). Thus, let a piecewise linear function with supporting points $r \in R = \{1, \dots, \lvert R \rvert\}$ approximating f determined by the procedure in Section 4 be given. Let the supporting points be $(\bar{\tau}_r, \bar{F}_r(\bar{\tau}_r))$, $r \in R$, i.e. $\bar{\tau}_r$ is the point in time of the supporting point and $\bar{F}_r(\bar{\tau}_r)$ the SOC. By this, the unique linear function cutting the supporting points $(\bar{\tau}_r, \bar{F}_r(\bar{\tau}_r))$ and $(\bar{\tau}_{r+1}, \bar{F}_{r+1}(\bar{\tau}_{r+1}))$ is $\bar{F}_r(\hat{t}) = \bar{b}_r + \bar{\beta}_r \cdot \hat{t}$ with $\hat{t}$ the time, $\bar{\beta}_r = \frac{\bar{F}(\bar{\tau}_{r+1}) - \bar{F}(\bar{\tau}_r)}{\bar{\tau}_{r+1} - \bar{\tau}_r}$ the slope, and $\bar{b}_r = \bar{F}(\bar{\tau}_r) - \bar{\beta}_r \cdot \bar{\tau}_r$ the intercept. Fig. 6 extends the piecewise linear function of Fig. 3(b) accordingly. We now describe a variation of the linearization presented by Zuo et al. (2019). The linearization by Zuo et al. (2019) has the benefit to require comparably few additional binary variables. In distinction to Zuo et al. (2019), we include the overestimation of the real charging function into the linearization. Moreover, we directly include the charging station variables y_{ic} in the linearization.

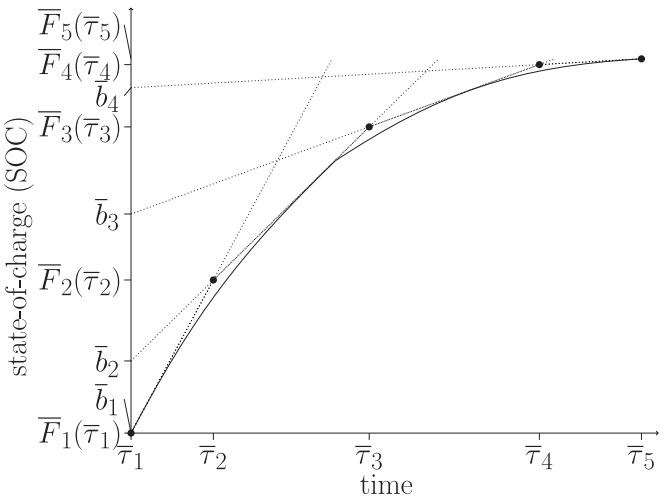


Fig. 6. Piecewise linear approximation of the charging function with five supporting points (overestimation).

We compute the time $\underline{t}_i$ to charge from a SOC of 0 to a SOC of $s_i - q_{ic} \cdot y_{ic}$ and the time $\bar{t}_i$ to charge from a SOC of 0 to a SOC of $s_i - q_{ic} \cdot y_{ic} + \phi_i$. Thus, $\bar{t}_i - \underline{t}_i$ is the required charging time and we can replace Constraints (11) by

$$z_i \geq \bar{t}_i - \underline{t}_i - \tilde{M}_i \cdot \left(1 - \sum_{c \in C} y_{ic}\right) \qquad \forall i \in I. \tag{18}$$

We first compute $\bar{t}_i$. As an example let us assume that $\overline{F}_3(\overline{\tau}_3) \leq s_i - q_{ic} \cdot y_{ic} + \phi_i \leq \overline{F}_4(\overline{\tau}_4)$ in Fig. 6. Thus, we get $\bar{t}_i$ by resolving the equation $\bar{b}_r + \bar{\beta}_r \cdot \bar{t}_i = s_i - q_{ic} \cdot y_{ic} + \phi_i$ with $r = 3$. However, in general we do not know r. Nevertheless, in general

$$s_i - q_{ic} \cdot y_{ic} + \phi_i \leq \bar{b}_r + \bar{\beta}_r \cdot \bar{t}_i + (1 - y_{ic}) \qquad \forall i \in I, c \in C, r = 1, \dots, |R| - 1 \tag{19}$$

holds.

To understand why (19) is correct, we go back to our example. Let $r < 3$, then $\bar{\beta}_r > \bar{\beta}_3$ and, although $\bar{b}_r < \bar{b}_3$, Fig. 6 shows that $\overline{F}_r(\bar{t}_i) = \bar{b}_r + \bar{\beta}_r \cdot \bar{t}_i \geq s_i - q_{ic} \cdot y_{ic} + \phi_i$ holds. Accordingly, Fig. 6 shows that this also holds for $r > 3$ due to the larger intercept $\bar{b}_r > \bar{b}_3$ which overcompensates for the smaller slope $\bar{\beta}_r < \bar{\beta}_3$. As this is also true in general for an arbitrary $r \in R$, Constraints (19) are a valid lower bound for $\bar{t}_i$. Because of (18) and since z_i is minimized in objective function (3), $\bar{t}_i$ is set to the correct value.

When computing $\underline{t}_i$ we cannot use the same trick because $\underline{t}_i$ is subtracted in (18) such that the model would maximize it. Instead, we use binary variables α_{ir} to identify the correct part of the piecewise linear charging function (the one for which $\alpha_{ir} = 1$ for the corresponding $i \in I$). Moreover, we are not allowed to underestimate $\underline{t}_i$. If we would do so, a too small number would be subtracted in (18) such that z_i could be overestimated. By this, a feasible solution can become infeasible due to the time windows. Hence, we cannot use $\bar{b}_r$ and $\bar{\beta}_r$, $r = 1, \dots, |R| - 1$, for the approximation. Instead, we need a piecewise linear function which overestimates the charging time (but underestimates the charging function) like in Fig. 2. We assume for now that a corresponding approximation $\underline{F}_r(\hat{t}) = \underline{b}_r + \underline{\beta}_r \cdot \hat{t}$, $r = 1, \dots, |R| - 1$, with supporting points $(\underline{\tau}_r, \underline{F}_r(\underline{\tau}_r))$, $r = 1, \dots, |R|$, is given. As the procedure is very similar to the approach in Section 4, we present the procedure to find them in the appendix.

Variables α_{ir} are set correctly by Constraints (20)–(23):

$$s_i - q_{ic} \cdot y_{ic} \geq (\underline{b}_r + \underline{\beta}_r \cdot \underline{\tau}_r) \cdot y_{ic} + (\alpha_{ir} - 1) \quad \forall i \in I, c \in C, r = 1, \dots, |R| - 1 \tag{20}$$

$$s_i - q_{ic} \cdot y_{ic} \leq (\underline{b}_{r+1} + \underline{\beta}_{r+1} \cdot \underline{\tau}_{r+1}) \cdot y_{ic} + (2 - \alpha_{ir} - y_{ic}) \quad \forall i \in I, c \in C, r = 1, \dots, |R| - 2 \tag{21}$$

$$\sum_{r \in R \setminus \{|R|\}} \alpha_{ir} = \sum_{c \in C} y_{ic} \quad \forall i \in I \tag{22}$$

$$\alpha_{ir} \in \{0, 1\} \quad \forall i \in I, r \in R \setminus \{|R|\} \tag{23}$$

$\underline{b}_r + \underline{\beta}_r \cdot \underline{\tau}_r$ is either strictly smaller or larger than $s_i - q_{ic} \cdot y_{ic}$ for all r beside the one we search for (if $s_i - q_{ic} \cdot y_{ic} = \underline{F}_{r'}(\underline{\tau}_{r'})$ for any $r' \in R \setminus \{|R|\}$, then r is not unique, but we are indifferent which of the two possible to choose). Moreover, the SOC is always smaller or equal to 1 such that (20) and (21) are always valid if $\alpha_{ir} = 0$. Thus, (20)–(23) give us the correct part of the piecewise linear function if $y_{ic} = 1$. If $y_{ic} = 0$, all constraints are fulfilled for $\alpha_{ir} = 0$ and for all $r = 1, \dots, |R| - 1$.

It remains to set $\underline{t}_i$ to the time to charge from a SOC of 0 to a SOC of $s_i - q_{ic} \cdot y_{ic}$. This is done by Constraints (24)–(25):

$$s_i - q_{ic} \cdot y_{ic} \geq \underline{b}_r + \underline{\beta}_r \cdot \underline{t}_i + (\alpha_{ir} - 1) \qquad \forall i \in I, c \in C, r = 1, \dots, |R| - 1 \tag{24}$$

$$s_i - q_{ic} \cdot y_{ic} \leq \underline{b}_r + \underline{\beta}_r \cdot \underline{t}_i + (2 - y_{ic} - \alpha_{ir}) \qquad \forall i \in I, c \in C, r = 1, \dots, |R| - 1 \tag{25}$$

Together, Constraints (18)–(25) linearize Constraints (11) by introducing $|I| \cdot (|R| - 1)$ further binary variables.

Remember the argumentation in Section 4 that a larger number of supporting points leads to a better approximation and therefore probably less cuts which need to be added to declare solutions to be infeasible. Now we are aware of the disadvantage of a larger number of supporting points leading to a larger number of binary variables. Thus, we have a trade-off between a better approximation and fewer binary variables. We evaluate this trade-off in the computational study.

5.3. Preprocessing

It is well-known that Big Ms should be set as small as possible (Camm et al., 1990; Codato and Fischetti, 2006). Therefore, we set

$$\begin{aligned}
M_{ij} &= l_i + t_{ij} - e_j \\
\tilde{M}_i &= \max\left\{ \max_{h \neq i}\{l_h - \min_{c \in C}\{t_{ic} + t_{ch}\}\} - e_i,\ \max_{c \in C, r \in R}\{\hat{t} : F_r(\hat{t}) = q_{c,n+1}\} \right\} \\
\bar{M}_{ijc} &= \max\left\{ 0, \max\left\{ \max_{h \neq i}\{l_h - \min_{c' \in C}\{t_{ic'} + t_{c'h}\}\} - e_i, \right.\right. \\
&\qquad \left.\left. \max_{c' \in C, r \in R}\{\hat{t} : F_r(\hat{t}) = q_{c',n+1}\} \right\} + l_i + t_{ic} + t_{cj} - e_j \right\}
\end{aligned}$$

whereat $\max_{h \neq i}\{l_h - \min_{c \in C}\{t_{ic} + t_{ch}\}\} - e_i$ is the longest time which can be charged at a charging station after location i if another location h is visited afterwards, i.e. the vehicle starts as early as possible in i, drives to the charging station c such that the travel time for the path $i \to c \to h$ is as short as possible, and reaches location h at the latest possible point in time. If the vehicle drives back to the depot after visiting location i, it will due to the objective never charge more than necessary to reach the depot. As we do not know which charging station is used, $\max_{c \in C, r \in R}\{\hat{t} : F_r(\hat{t}) = q_{c0}\}$ is an upper bound for the charging time in this case. Note that the charging time has to be computed by the real charging function here. As the piecewise linear approximation overestimates the charging function, it underestimates the required charging time. Thus, a feasible solution could be excluded by the constraint if the approximation is used. As either another location or the depot has to be visited after the charging procedure, we would never charge for a longer time than the maximum of both. To do so would either be infeasible or not optimal due to an unnecessary high charging time. Beside that all three choices of Big M are set in the classical way as difference between the largest value the right side of the inequality can obtain and the smallest value the left side of the inequality can obtain. By this, the constraints are fulfilled for all optimal solutions.

Moreover, we can fix x_{ij} variables to 0 if they cannot be selected in a feasible tour due to time windows (Cordeau, 2006). This is the case if the vehicle would not reach location j within its time window even if it starts in location i at the beginning of location i's time window. Thus, x_{ij} can be fixed to 0 if

$$e_i + t_{ij} > l_j.$$

With the same argument we know that charging station c cannot be visited between locations i and j if

$$e_i + t_{ic} + t_{cj} > l_j.$$

If this is the case, we can add the following cut to the model

$$y_{ic} + x_{ij} \leq 1. \tag{26}$$

We cannot fix y_{ic} directly to 0, as there might be another location h such that $i \to c \to h$ is a feasible sequence. However, if this sequence is infeasible for all $h \in I \backslash \{0, n+1\}$, Constraints (26) ensure already that $y_{ic} = 0$ if another customer location is visited after i. Though, the sequence $i \to c \to n+1$ is still possible.

Keskin and Çatay (2018) stated that a charging station will not be visited between locations i and j if there is another charging station which is closer to location i and to location j. Transmitted to our setting a charging station is dominated to be visited between locations i and j if there is another non-dominated charging station c such that the vehicle requires less energy driving from location i to c as well as from there to location j in comparison to the corresponding segments for the dominated charging station. For all dominated charging stations (26) is a valid inequality.

5.4. Separation procedure

Our piecewise linear approximation of the charging function underestimates the required charging time such that it is possible that the vehicle charges more energy than possible in the corresponding time. If the vehicle reaches a location at the end of the time window, this might lead to an infeasible solution because, using the real charging function, not enough energy can be charged. To identify such infeasibilities, we run through all tours in an integer solution and check them. To do so, we start in the depot and check Constraints (9)–(14). If the time window at location j or a later location is violated, we cannot charge ϕ_i at the charging station visited after the previous location i. Then, we check whether it is possible to push up charge to a charging station visited prior in the tour. If so, we charge such that j is again reached with a SOC of s_j or if this is not possible as much as possible such that no time window is violated and we reach the next charging stop after j as early as possible. Due to the concave charging function it is optimal to charge as late as possible. Thus, it is better to reach the next stop earlier than charging more before location j. From there on we proceed to check the remaining tour. If we cannot charge enough at any charging stop to reach the next charging stop or the end depot at least with a SOC of 0, the tour is infeasible. Let k be the next location visited after the charging stop (or the end depot) which cannot be reached without the battery being depleted. Then, we add the following cut and truncate the node:

$$\sum_{(i,j)\in S_k} x_{ij} + \sum_{(i,c)\in \bar{S}_k} y_{ic} \leq |S_k| + |\bar{S}_k| - 1 + \sum_{(i,c)\in \hat{S}_k} y_{ic} \tag{27}$$

whereat (i,j) is in S_k if the vehicle visits location j directly or via a charging station after location i and before location k in the infeasible tour. If k is the end depot, $(i, n+1)$ is included in S_k for the location i visited directly before the end depot. Accordingly, y_{ic} is in $\bar{S}_k$ if the vehicle visits charging stop c directly after location i and location i is visited before location k in the infeasible tour. Set $\hat{S}_k$ contains all pairs of locations visited in the tour before location k and charging stops c such that $y_{ic} = 0$ in the current solution. As the tour segment $0 \to \cdots \to c$ where c is the charging stop directly before location k (and accordingly tour $0 \to \cdots \to n+1$) is infeasible, at least one of the corresponding variables cannot be 1 in a feasible solution or an additional charging station needs to be visited such that (27) cuts of the infeasible solution.

If a tour is feasible, we compare the computed charging time Z as described above with the required charging time in the solution $\sum_{i\in S} z_i$ whereat S is the set of all customer locations visited in the tour. If $Z > \sum_{i\in S} z_i$, the real charging time is underestimated such that the solution is not evaluated correctly. To correct this, we add the following cut

$$\sum_{i\in S} z_i \geq Z \cdot \left(\sum_{(i,j)\in S_x} x_{ij} + \sum_{(i,c)\in S_y} y_{ic} - (|S_x| + |S_y| - 1) - \sum_{i\in S} \sum_{(i,c)\notin S_y} y_{ic} \right) \tag{28}$$

whereat S_x is the set of pairs of locations (i,j) (including depot) such that location j is visited directly or via a charging stop after location i in the tour and S_y is the set of all pairs (i,c) of a location i and a charging station c such that c is visited directly after location i in the tour. Thus, the term in brackets is 1 if exactly the considered tour is selected. If anything in the tour changes, i.e. either $\sum_{(i,j)\in S_x} x_{ij} + \sum_{(i,c)\in S_y} y_{ic}$ is reduced by at least 1 or an additional stop is added ($\sum_{i\in S} \sum_{(i,c)\notin S_y} y_{ic}$ is increased by at least 1), the term in the brackets is at most 0. Thus, (28) is a valid inequality.

6. Computational study

The objective of the computational study is to evaluate the performance of the presented approach as well as the trade-off between an increased number of binary variables for an increased number of supporting points (compare Section 5.2) and a worse approximation of the linearization for a smaller number of supporting points. The experiments were conducted on a single AMD EPYC 7542 32 core with 2.90 GHz. The time limit for each instance was set to 3600s. The composition is described in Section 6.1. The results are presented in Section 6.2.

6.1. Composition

We use the following non-linear charging function model introduced by Motoaki et al. (2018) for the computational study:

$$f(\hat{t}) = \frac{0.015 + 0.00034 \cdot T}{0.022} - \frac{0.015 + 0.00034 \cdot T}{0.022} \cdot e^{-0.022 \cdot \hat{t}} \tag{29}$$

whereat T is the outside temperature. As $\lim_{\hat{t}\to\infty} e^{-0.022\cdot\hat{t}} = 0$, $f(\hat{t}) = 1$ can only be reached if $\frac{0.015+0.00034\cdot T}{0.022} \geq 1$, i.e. $T \geq \frac{0.007}{0.00034} \approx 20.59$. In our experiments, we set $T = 25$, which is a realistic temperature in many countries, such that $f(\hat{t}) = 1$ for $\hat{t} = -\frac{\ln\left(1 - \frac{0.022}{0.015+0.00034\cdot 25}\right)}{0.022} \approx 125.07$. This means a vehicle requires 125.07 time units to fully recharge an empty battery. We approximated Eq. (29) with three, five, and seven supporting points. The resulting linear approximations, determined with the procedures described in Section 4 and in the appendix, can be found in Table 2. The error is the percentage of the value in (1) and (30), respectively, in relation to $\int_0^{f^{-1}(1)} f(\hat{t})d\hat{t}$. Due to the convexity of $f(\cdot)$ the error is smaller for the piecewise linear approximation overestimating $f(\cdot)$. We set $\varepsilon = \bar{\varepsilon} = 0.001$ when determining the piecewise linear charging functions.

We evaluated instances with 10, 15, 20, 30, 40, 50, 80, and 100 customer locations as well as 2, 5, and 10 charging stations. The number of vehicles K was set dependent on the number of customer locations such that on average 5 ($K = \lceil n/5 \rceil$), 10 ($K = \lceil n/10 \rceil$), and 20 ($K = \lceil n/20 \rceil$) locations are visited in one tour. Moreover, we considered three different time window length $TW length$ of 0, 10, and 20 time units. Finally, we determined three instances for each possible combination resulting from the instance characteristics described above. In total, we get 621 instances (for 10 customers locations $\lceil n/10 \rceil = 1 = \lceil n/20 \rceil$).

Table 2
Linear approximations of the charging function (rounded values).

	Three supporting points (error over: 3.34%, error under: 6.85%)					
	$\overline{\tau}_r$	$\overline{b}_r$	$\overline{\beta}_r$	$\underline{\tau}_r$	$\underline{b}_r$	$\underline{\beta}_r$
1	0	0.1091	0.0137	0	0	0.0144
2	48.772	0.6086	0.0034	48.855	0.5135	0.0039
3	125.07	–	–	125.07	–	–
	Five supporting points (error over: 0.84%, error under: 1.69%)					
	$\overline{\tau}_r$	$\overline{b}_r$	$\overline{\beta}_r$	$\underline{\tau}_r$	$\underline{b}_r$	$\underline{\beta}_r$
1	0	0.0292	0.0182	0	0	0.0186
2	22.934	0.2063	0.0105	22.167	0.1687	0.0110
3	49.888	0.4583	0.0055	48.172	0.4195	0.0058
4	82.586	0.7110	0.0024	81.327	0.6846	0.0025
5	125.07	–	–	125.07	–	–
	Seven supporting points (error over: 0.38%, error under: 0.76%)					
	$\overline{\tau}_r$	$\overline{b}_r$	$\overline{\beta}_r$	$\underline{\tau}_r$	$\underline{b}_r$	$\underline{\beta}_r$
1	0	0.0148	0.0197	0	0	0.0200
2	15.890	0.1113	0.0136	15.095	0.0904	0.0140
3	33.294	0.2612	0.0091	32.487	0.2425	0.0093
4	52.252	0.4303	0.0059	52.246	0.4210	0.0059
5	73.148	0.5968	0.0036	74.135	0.5911	0.0036
6	96.860	0.7483	0.0020	97.012	0.7405	0.0021
7	125.07	–	–	125.07	–	–

Instances were determined as follows: Locations were drawn randomly on a 100×100 grid. For it, we drew a random position on both axes for each location (customer locations, depot, charging stations) according to a uniform distribution. Afterwards, travel times were determined according to the Euclidean distance metric. This value was then divided by the battery capacity to obtain the required energy. To obtain a sufficient driving range we set the battery capacity to two times the grid's diagonal.

It remains to determine the time windows. However, the time windows are critical for feasibility. To ensure that a feasible solution exists, we determined a feasible solution. For it, a random not yet assigned location is selected and assigned randomly to one of the tours until all locations are assigned. For each tour the customer locations are served in the sequence they were assigned to the tour. Next, we need to include the charging stops. To do so, we start at the depot and run through the tour until the battery is depleted. Let sum be the sum of required energy until the current location. We sum up $1/(1-sum)$ at each location. Let this sum until the current location be called a and the same sum until the last location before the battery would be depleted be called b. Afterwards, a random number in $[0, 1]$ is determined and the charging stop is scheduled after the location where a/b exceeds the random number for the first time. By this, it is more likely that the charging stop is scheduled when the SOC is lower. Which of the reachable charging stations is used is again determined according to a uniform distribution. Afterwards, the same procedure is repeated starting in the newly scheduled charging stop until the end depot is reached. This was done for all tours. Finally, the time windows were determined by running through each tour, including recharging up to the required amount to travel to the next stop at each selected charging station. Thereby, charging times were determined according to function (29). Then, e_i was set to the time location i is reached, which is the earliest possible time given this tour, and $l_i = e_i + TWlength$. Note that due to the randomly determined tours it is very likely that there are several feasible solutions even if $TWlength = 0$.

6.2. Results

We first evaluate the performance of the presented approach before the mentioned trade-off is discussed. Our instances vary in five dimensions: the number of vehicles, the number of charging stations, the average number of customers per vehicle (called workload), the time window length ($TWlength$), and the number of supporting points. We evaluate all criteria in dependency of the first, the number of customer locations.

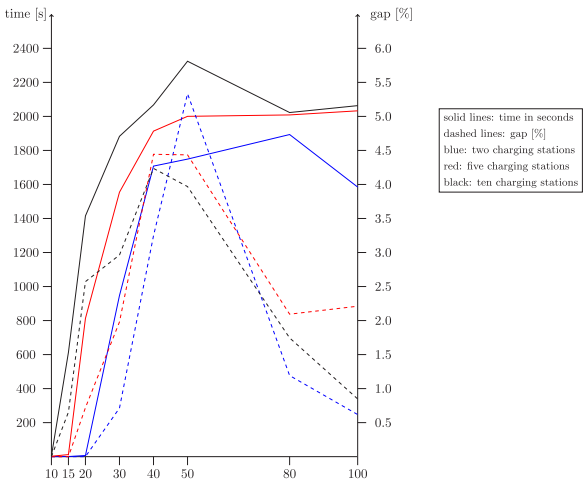


Fig. 7. Evaluation of solutions regarding number of charging stations.

6.2.1. Evaluation of performance

Fig. 7 shows the average computation times of all instances with the corresponding number of customer locations and charging stations (solid lines) as well as their average gaps (dashed lines). Regarding the computation times it can clearly be seen that computation times are the higher the larger the number of possible charging stations is. For the gaps the impression is not that clear. Although the gaps after one hour follow the same profile – increasing up to 40 or 50 customer locations and decreasing afterwards –, there is no clear tendency whether a larger or smaller number of charging stations leads to better or worse results. However, an interesting observation is that gaps have a peak for 40 or 50 customer locations. A reason for it might be the interaction between two effects. First, a smaller number of customer locations leads to easier instances due to less binary variables. Second, as instances are determined randomly, it is more likely that similar customer locations (position on the grid and time windows) are drawn if the number of customer locations is higher such that it is easier to find good solutions for a larger number of customer locations. Note, furthermore, that we included a dominance criterion for the charging stations (compare Section 5.3). Thus, the number of non-dominated charging stations might be significantly lower than the number of possible charging stations. Overall, the number of possible charging stations seems to have a rather small effect on the instances' difficulty.

This impression is confirmed by the evaluation of the vehicles' workload in Fig. 8. As before the average computation times (solid lines) and the average gaps (dashed lines) are presented for the three different average numbers of customer locations visited by each vehicle. Note here that we do not have instances with 10 customer locations and an average of 20 customers per vehicle. Fig. 8 clearly shows that a larger workload per vehicle leads to more difficult instances. A reason might be the aforementioned effect that it is more likely to have similar customer locations the larger the number of customer locations is. If the number of customer locations per vehicle is smaller, it is of course more likely that we find a group of customer locations of the required size which are concentrated locally on the grid and have fitting time windows. The figure also shows that our branch-and-cut approach struggles with instances with 50 or more customer locations and an average of 20 customers per vehicle. We were not able to solve these instances within one hour of computation time. Nevertheless, instances with fewer customers per vehicle were solved reliably even for instances with 100 customer locations.

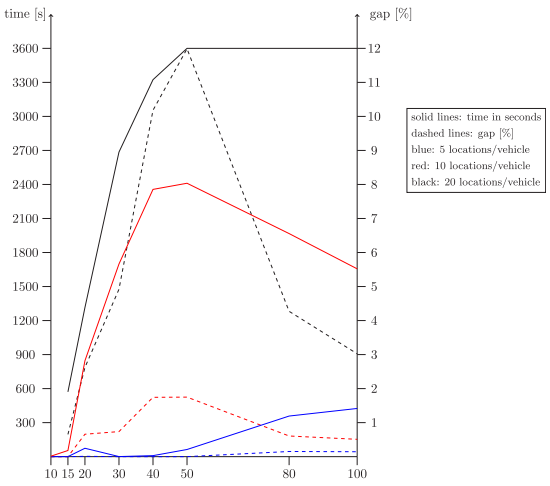


Fig. 8. Evaluation of solutions regarding workload of vehicles.

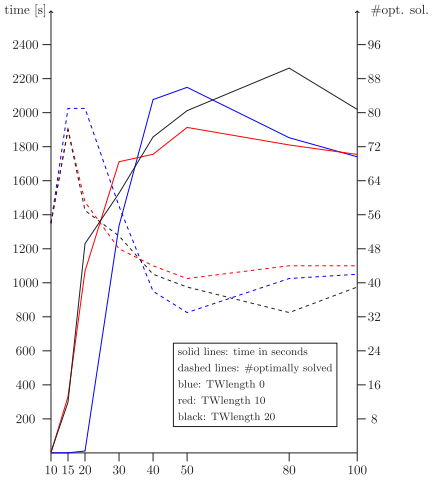


Fig. 9. Evaluation of solutions regarding time window length.

It is well-known in the routing literature that larger time windows typically increase the instance's difficulty, as smaller time windows reduce the number of feasible tours. We also fixed variables to 0 if we know already that they cannot be part of a feasible solution due to time windows (compare Section 5.3). Fig. 9 analyzes the performance regarding the different time window lengths. Solid lines show the average computation times again. Dashed lines present the number of instances terminated with a proven optimal solution (there are 81 instances with fixed time window length for each number of customer locations; 54 for ten customer locations). In our results, we can only observe a small effect of the time window size on the solution quality. For 80 and 100 customer locations the instances with a time window lengths of 20 led to the longest computation times and the fewest instances were solved to proven optimality. However, the figure confirms the results obtained in Fig. 7 that instances are more difficult if more customer locations are present. Moreover, we can see that almost all instances were solved to proven optimality for a small number of customer locations (especially 10 and 15). Afterwards, this number decreases. Starting with 40 or 50 locations roughly half of the instances were solved to proven optimality. Remembering the results in regard to the vehicles' workload, these are especially those instances with a small workload per vehicle, i.e. with a larger number of vehicles.

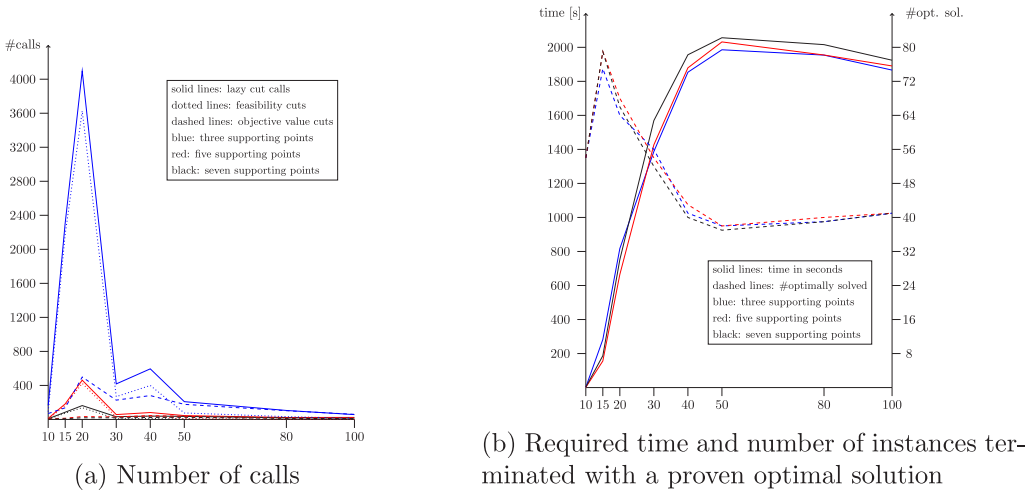
Concluding the evaluation of our solution approach, we can summarize that our branch-and-cut framework is able to solve even large instances with 100 customer locations to proven optimality. If the number of vehicles is not too large, even almost all of these instances can be solved to optimality within one hour of computation time. Otherwise, gaps of a few percent are obtained. Moreover, the workload of vehicles has the largest effect on the solution quality while the number of charging stations and the time window length only have a smaller influence.

6.2.2. *Influence of the number of supporting points*

Our solution approach has the characteristic to allow infeasible solutions in the model formulation which are declared to be infeasible by an infeasible path cut if they occur. By this, we are able to solve the linearized model without cutting off feasible solutions. Fig. 10(a) pictures the number of calls of the procedure described in Section 5.4 (solid lines), i.e. how often integer solutions were considered in the branch-and-cut procedure. Dashed lines represent the number of cuts of type (28) added to the model. Dotted lines do the same for cuts of type (27). It can clearly be seen that in most calls infeasible path cuts were added to the model if the number of customer locations is rather small (up to 40). Note that only changing the charging station visited between two customer locations leads to a new integer solution which explains the significant number of calls in instances with only three supporting points. Nevertheless, 4000 is still a moderate number such that the required computation time for the procedure in Section 5.4 is negligible in comparison to the computation time of the entire branch-and-cut procedure. If the number of customer locations becomes larger, the number of added cuts of type (27) decreases. Instead, most of the found solutions were feasible, but the required charging time needed to be corrected. Furthermore, the left part of Fig. 10 shows that the number of added cuts can be significantly reduced by increasing the number of supporting points. However, the right side of the figure shows that the different numbers of supporting points led to very similar results regarding the required computation times as well as the number of instances solved to proven optimality. Thus, the results show that the effect of a higher model complexity due to more binary variables in favor of a better approximation of the charging function is comparable to the effect of a higher chance to obtain integer solutions which need to be declared as infeasible if the number of supporting points and by it the number of binary variables is smaller, i.e. the charging function approximation is worse. In tendency, it seems to be that seven supporting points are already too many, as for instances with at least 30 customer locations fewer optimal solution were found and higher computation times were required in comparison with the same instances with three and five supporting points, respectively. Note that we used the same number of supporting points for the piecewise linear function overestimating the real charging function as well as for the one underestimating the real charging function. However, additional binary variables are only required to determine $\underline{t}_i$ in Constraints (20)–(25). The approximation of $\bar{t}_i$ (in Constraints (19)) can be improved by further supporting points by only adding $|I| \cdot |C|$ additional constraints per supporting point. Nevertheless, Fig. 10 shows that the number of supporting points is not really decisive. Therefore, we abstain from evaluating different numbers of supporting points for the two approximations.

6.2.3. *Relation between optimal and best truncated solution*

Our procedure, moreover, allows us to track the best solution which was truncated by an infeasible path cut (27) for every instance. Fig. 11 shows the relative improvement of the best truncated solution in relation to the optimal solution (best feasible found if procedure not terminated within 3600s). Solid lines show the average over all instances with the corresponding number of customer locations and dashed lines the maximum of them. It can clearly be seen that a smaller number of supporting points leads to a higher difference between optimal solution and best truncated solution. This is not surprising, as the approximation is worse if the number of supporting points is smaller (compare Fig. 3). For three supporting points we found gaps



(a) Number of calls

(b) Required time and number of instances terminated with a proven optimal solution

Fig. 10. Influence of the number of supporting points on the solution quality.

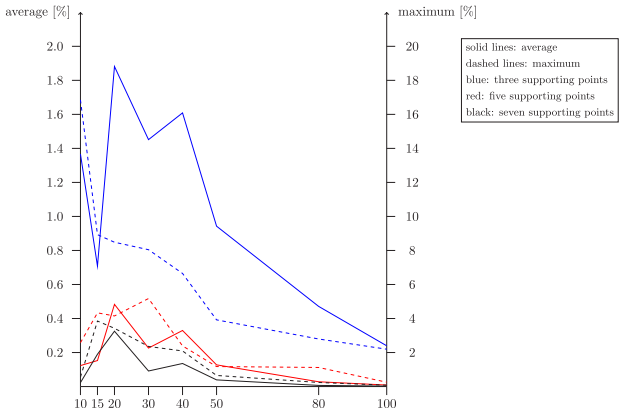


Fig. 11. Evaluation of truncated infeasible solutions in relation to optimal solution.

of up to almost 2% on average and over 16% in maximum. However, for five supporting points (5.17%) and seven supporting points (3.85%) maximal gaps were clearly smaller and average gaps were smaller than 0.5%.

We evaluated the gaps between piecewise linear overestimation of the real charging function and the real charging function. Thus, we cannot conclude on the gaps reached by an underestimation of the real charging function as it is typically done in the literature. However, Table 2 shows that the approximation error is rather smaller for the overestimation in comparison to the underestimation such that the results indicate that the approximation error with a piecewise linear underestimation might be larger than 0.5% for some instances even if more than three supporting points are used.

7. Conclusion

We have investigated the EVRP with a realistic non-linear concave and strictly monotonic increasing charging function. In our model, we approximate the charging function by a piecewise linear function which never underestimates the real SOC. By this, the solution space is increased. Whenever an integer solution is found in the branch-and-cut procedure, we declare infeasible solutions regarding the real charging function to be infeasible by adding an infeasible path cut of type (27). If the solution is feasible, we correct the required charging time if necessary by adding a cut of type (28). The computational results show that the approach can solve instances with 100 customer locations reliably to optimality. Moreover, we found that the average number of customers per vehicle is the main indicator for instances to be difficult to solve. The number of charging stations as well as the time window length had smaller effects in this regard. The larger the number of supporting points is, the better is the approximation, but the larger is the number of binary variables in the model formulation. Our computational results show that there is no clear difference in the performance for three, five, and seven supporting points.

As the presented solution approach leads to good results even for larger instances with 100 customer locations, it can be used in different routing settings in future research to incorporate the usage of electric vehicles. Examples are variants of the EVRP e.g. with vehicle capacities or maximum ride time constraints. Moreover, possible applications also include the pickup-and-delivery problem or the dial-a-ride problem and variants of them like ridepooling services (Schulz and Pfeiffer, 2024). In applications of all of these problems, electric vehicles are used for emission-free transportation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Piecewise linear charging function underestimating real charging function

As described in Section 5.2 we also need a piecewise linear approximation of the charging function such that the real charging function is never overestimated, but the estimation is as tight as possible. This means, we want to find linear functions $\underline{F}_r(\hat{t}) = \underline{b}_r + \underline{\beta}_r \cdot \hat{t}$, $r = 1, \dots, |R|-1$, and $\underline{F}(\hat{t}) = \min_{r \in R \setminus \{|R|\}} \underline{F}_r(\hat{t})$ such that $\underline{F}(\hat{t}) \leq f(\hat{t})$ holds for all $\hat{t} \in [0, f^{-1}(1)]$ and

$$\int_0^{f^{-1}(1)} f(\hat{t})d\hat{t} - \int_0^{f^{-1}(1)} \underline{F}(\hat{t})d\hat{t} \tag{30}$$

is minimized. Analogously to (1), which is minimized if Eq. (2) is minimized, (30) is minimized if

$$\int_0^{f^{-1}(1)} \underline{F}(\hat{t})d\hat{t} = \sum_{r=1}^{|R|-1} \frac{1}{2} \cdot (\underline{F}_{r+1}(\underline{\tau}_{r+1}) - \underline{F}_r(\underline{\tau}_r)) \cdot (\underline{\tau}_{r+1} - \underline{\tau}_r) + \underline{F}_r(\underline{\tau}_r) \cdot (\underline{\tau}_{r+1} - \underline{\tau}_r) \tag{31}$$

is maximized. Analogously to the procedure in Section 4, we first find a starting solution by splitting the interval $[0, f^{-1}(1)]$ into $|R|$ equally long intervals and setting $\underline{\tau}_1 = 0$, $\underline{\tau}_r$, $1 < r < |R|$, as the intersection of two succeeding intervals, and $\underline{\tau}_{|R|} = f^{-1}(1)$. Then, $\underline{F}_r$ is defined as the unique linear function crossing $f(\underline{\tau}_r)$ and $f(\underline{\tau}_{r+1})$, $r = 1, \dots, |R| - 1$.

Afterwards, we use Algorithm 2 to optimize this solution but run the loop in line 5 from $r = 2$ to $|R| - 1$ because the first supporting point is always $(0, 0)$ and call Algorithm 3 instead of Algorithm 1 in line 6. If $r = |R| - 1$, Algorithm 3 requires $\underline{F}_{|R|}(\cdot)$. As the last supporting point is always $(f^{-1}(1), 1)$, we can simply set $F_{|R|}(\cdot) = 1$.

Algorithm 3. *(Determination of optimal supporting points (underestimation))*

1: ***function*** SET_SUPPORTING_POINT($\underline{\tau}_{r-1}$, $\underline{\tau}_r$, $\underline{\tau}_{r+1}$, $f(\cdot)$, $\underline{F}_{r-1}(\cdot)$, $\underline{F}_{r+1}(\cdot)$, ε)
2: *let $\hat{p}$ be the increasingly ordered list of possible supporting points $\hat{p} = \left(\frac{\underline{\tau}_{r-1}+\underline{\tau}_r}{2}, \underline{\tau}_r, \frac{\underline{\tau}_r+\underline{\tau}_{r+1}}{2}\right)$*
3: *set $g(1) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\underline{F}_{r'+1}(\underline{\tau}_{r'+1}) - \underline{F}_{r'}(\underline{\tau}_{r'})) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'}) + \underline{F}_{r'}(\underline{\tau}_{r'}) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'})$ whereat $\underline{F}_r(\cdot)$ is the unique linear function determined by $\frac{\underline{\tau}_{r-1}+\underline{\tau}_r}{2}$, $f\left(\frac{\underline{\tau}_{r-1}+\underline{\tau}_r}{2}\right)$, $\underline{\tau}_{r+1}$, and $f(\underline{\tau}_{r+1})$*
4: *set $g(2) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\underline{F}_{r'+1}(\underline{\tau}_{r'+1}) - \underline{F}_{r'}(\underline{\tau}_{r'})) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'}) + \underline{F}_{r'}(\underline{\tau}_{r'}) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'})$ whereat $\underline{F}_r(\cdot)$ does not change*
5: *set $g(3) = \sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\underline{F}_{r'+1}(\underline{\tau}_{r'+1}) - \underline{F}_{r'}(\underline{\tau}_{r'})) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'}) + \underline{F}_{r'}(\underline{\tau}_{r'}) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'})$ whereat $\underline{F}_r(\cdot)$ is the unique linear function determined by $\frac{\underline{\tau}_r+\underline{\tau}_{r+1}}{2}$, $f\left(\frac{\underline{\tau}_r+\underline{\tau}_{r+1}}{2}\right)$, $\underline{\tau}_{r+1}$, and $f(\underline{\tau}_{r+1})$*
6: *set current = 0*
7: *set best = $\min\{g(1), g(2), g(3)\}$*
8: ***while*** *best − current > ε* ***do***
9: *set best = current*
10: *$u = 1$*
11: ***while*** *$g(u) > g(u+1)$ and $u < |\hat{p}|$* ***do***
12: *set $u = u + 1$*
13: ***end while***
14: *insert $\underline{\tau}_r = \frac{\hat{p}_u+\hat{p}_{u+1}}{2}$ in list $\hat{p}$ ($\hat{p}_u$ is the $(u+1)^{th}$ entry in extended list $\hat{p}$), evaluate the solution ($\sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\underline{F}_{r'+1}(\underline{\tau}_{r'+1}) - \underline{F}_{r'}(\underline{\tau}_{r'})) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'}) + \underline{F}_{r'}(\underline{\tau}_{r'}) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'})$ whereat $F_r(\cdot)$ is the unique linear function determined by $\frac{\hat{p}_u+\hat{p}_{u+1}}{2}$, $f\left(\frac{\hat{p}_u+\hat{p}_{u+1}}{2}\right)$, $\underline{\tau}_{r+1}$, and $f(\underline{\tau}_{r+1})$), update $g(\cdot)$, and update current if improved*
15: *set $u = |\hat{p}|$*
16: ***while*** *$g(u-1) < g(u)$ and $u > 0$* ***do***
17: *set $u = u - 1$*
18: ***end while***
19: *insert $\underline{\tau}_r = \frac{\hat{p}_{u-1}+\hat{p}_u}{2}$ in list $\hat{p}$ ($\hat{p}_u$ is the $(|\hat{p}| - u - 1)^{th}$ entry in extended list $\hat{p}$), evaluate the solution ($\sum_{r'=r-1}^{r} \frac{1}{2} \cdot (\underline{F}_{r'+1}(\underline{\tau}_{r'+1}) - \underline{F}_{r'}(\underline{\tau}_{r'})) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'}) + \underline{F}_{r'}(\underline{\tau}_{r'}) \cdot (\underline{\tau}_{r'+1} - \underline{\tau}_{r'})$ whereat $\underline{F}_r(\cdot)$ is the unique linear function determined by $\frac{\hat{p}_{u-1}+\hat{p}_u}{2}$, $f\left(\frac{\hat{p}_{u-1}+\hat{p}_u}{2}\right)$, $\underline{\tau}_{r+1}$, and $f(\underline{\tau}_{r+1})$), update $g(\cdot)$, and update current if improved*
20: ***end while***
21: ***return*** *$\underline{\tau}_r$ and $F_r(\cdot)$ determined by $p = arg\ min_{p' \in \hat{p}}\{g(p')\}$.*
22: ***end function***

References

Boysen, N., Briskorn, D., Schwerdfeger, S., 2023. How to charge while driving: scheduling point-to-point deliveries of an electric vehicle under overhead wiring. J. Sched. 26, 19–41.

Camm, J.D., Raturi, A.S., Tsubakitani, S., 1990. Cutting big M down to size. Interfaces 20 (5), 61–66.

Camponogara, E., Nazari, L.F., 2015. Models and algorithms for optimal piecewise-linear function approximation. Math. Probl. Eng. 2015.

Ceselli, A., Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G., 2021. A branch-and-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies. In: Operations Research Forum. Vol. 2, Springer, pp. 1–33.

Codato, G., Fischetti, M., 2006. Combinatorial Benders' cuts for mixed-integer linear programming. Oper. Res. 54 (4), 756–766.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54 (3), 573–586.

Cortés-Murcia, D.L., Prodhon, C., Afsar, H.M., 2019. The electric vehicle routing problem with time windows, partial recharges and satellite customers. Transp. Res. E 130, 184–206.

Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. 2 (4), 393–410.

Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. 64 (6), 1388–1405.

Dönmez, S., Koç, Ç., Altıparmak, F., 2022. The mixed fleet vehicle routing problem with partial recharging by multiple chargers: Mathematical model and adaptive large neighborhood search. Transp. Res. E 167, 102917.

Duman, E.N., Taş, D., Çatay, B., 2022. Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows. Int. J. Prod. Res. 60 (17), 5332–5353.

Erdelić, T., Carić, T., 2019. A survey on the electric vehicle routing problem: variants and solution approaches. J. Adv. Transp. 2019.

Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transp. Res. E 48 (1), 100–114.

Froger, A., Jabali, O., Mendoza, J.E., Laporte, G., 2022. The electric vehicle routing problem with capacitated charging stations. Transp. Sci. 56 (2), 460–482.

Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294.

Ghorbani, E., Alinaghian, M., Gharehpetian, G.B., Mohammadi, S., Perboli, G., 2020. A survey on environmentally friendly vehicle routing problem and a proposal of its classification. Sustainability 12 (21), 9079.

Guo, F., Zhang, J., Huang, Z., Huang, W., 2022. Simultaneous charging station location-routing problem for electric vehicles: Effect of nonlinear partial charging and battery degradation. Energy 250, 123724.

Hiermann, G., Hartl, R.F., Puchinger, J., Vidal, T., 2019. Routing a mix of conventional, plug-in hybrid, and electric vehicles. European J. Oper. Res. 272 (1), 235–248.

Kancharla, S.R., Ramadurai, G., 2020. Electric vehicle routing problem with non-linear charging and load-dependent discharging. Expert Syst. Appl. 160, 113714.

Keskin, M., Çatay, B., 2018. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Comput. Oper. Res. 100, 172–188.

Keskin, M., Laporte, G., Çatay, B., 2019. Electric vehicle routing problem with time-dependent waiting times at recharging stations. Comput. Oper. Res. 107, 77–94.

Klein, P.S., Schiffer, M., 2022. Electric vehicle charge scheduling with flexible service operations. arXiv preprint arXiv:2201.03972.

Koç, Ç., Jabali, O., Mendoza, J.E., Laporte, G., 2019. The electric vehicle routing problem with shared charging stations. Int. Trans. Oper. Res. 26 (4), 1211–1243.

Kucukoglu, I., Dewil, R., Cattrysse, D., 2021. The electric vehicle routing problem and its variations: A literature review. Comput. Ind. Eng. 161, 107650.

Kullman, N.D., Goodson, J.C., Mendoza, J.E., 2021. Electric vehicle routing with public charging stations. Transp. Sci. 55 (3), 637–659.

Lam, E., Desaulniers, G., Stuckey, P.J., 2022. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. Comput. Oper. Res. 145, 105870.

Lee, C., 2021. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. J. Oper. Res. Soc. 72 (7), 1461–1485.

Lera-Romero, G., Miranda Bront, J.J., Soulignac, F.J., 2024. A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows. Eur. J. Oper. Res. 312 (3), 978–995.

Liang, Y., Dabia, S., Luo, Z., 2021. The electric vehicle routing problem with nonlinear charging functions. arXiv preprint arXiv:2108.01273.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G., 2019. The green mixed fleet vehicle routing problem with partial battery recharging and time windows. Comput. Oper. Res. 101, 183–199.

Marra, F., Yang, G.Y., Træholt, C., Larsen, E., Rasmussen, C.N., You, S., 2012. Demand profile study of battery electric vehicle under different charging options. In: 2012 IEEE Power and Energy Society General Meeting. IEEE, pp. 1–7.

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM 7 (4), 326–329.

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. B 103, 87–110.

Motoaki, Y., Yi, W., Salisbury, S., 2018. Empirical analysis of electric vehicle fast charging under cold temperatures. Energy Policy 122, 162–168.

Pelletier, S., Jabali, O., Laporte, G., 2016. 50Th anniversary invited article—goods distribution with electric vehicles: review and research perspectives. Transp. Sci. 50 (1), 3–22.

Pelletier, S., Jabali, O., Laporte, G., 2018. Charge scheduling for electric freight vehicles. Transp. Res. B 115, 246–269.

Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. Transp. Res. B 103, 158–187.

Qin, H., Su, X., Ren, T., Luo, Z., 2021. A review on the electric vehicle routing problems: Variants and algorithms. Front. Eng. Manage. 8, 370–389.

Raeesi, R., Zografos, K.G., 2022. Coordinated routing of electric commercial vehicles with intra-route recharging and en-route battery swapping. European J. Oper. Res. 301 (1), 82–109.

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

Scholl, J., Boysen, N., Scholl, A., 2023. E-platooning: Optimizing platoon formation for long-haul transportation with electric commercial vehicles. European J. Oper. Res. 304 (2), 525–542.

Schulz, A., Pfeiffer, C., 2024. Using fixed paths to improve branch-and-cut algorithms for precedence-constrained routing problems. European J. Oper. Res. 312 (2), 456–472.

Schulz, A., Suzuki, Y., 2023. An efficient heuristic for the fixed-route vehicle-refueling problem. Transp. Res. E 169, 102963.

Schwerdfeger, S., Bock, S., Boysen, N., Briskorn, D., 2022. Optimizing the electrification of roads with charge-while-drive technology. European J. Oper. Res. 299 (3), 1111–1127.

Su, Y., Dupin, N., Puchinger, J., 2023. A deterministic annealing local search for the electric autonomous dial-a-ride problem. European J. Oper. Res. 309 (3), 1091–1111.

Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications, second ed. In: MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, URL https://books.google.de/books?id=VUzUBQAAQBAJ.

Uhrig, M., Weiß, L., Suriyah, M., Leibfried, T., 2015. E-mobility in car parks–guidelines for charging infrastructure expansion planning and operation based on stochastic simulations. In: EVS28 International Electric Vehicle Symposium and Exhibition. pp. 1–12.

Verma, A., 2018. Electric vehicle routing problem with time windows, recharging stations and battery swapping stations. EURO J. Transp. Logist. 7 (4), 415–451.

Wu, Z., Zhang, J., 2021. A branch-and-price algorithm for two-echelon electric vehicle routing problem. Complex Intell. Syst. 1–16.

Xiao, Y., Zhang, Y., Kaku, I., Kang, R., Pan, X., 2021. Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption. Renew. Sustain. Energy Rev. 151, 111567.

Xu, M., Meng, Q., 2019. Fleet sizing for one-way electric carsharing services considering dynamic vehicle relocation and nonlinear charging profile. Transp. Res. B 128, 23–49.

Xu, W., Zhang, C., Cheng, M., Huang, Y., 2022. Electric vehicle routing problem with simultaneous pickup and delivery: Mathematical modeling and adaptive large neighborhood search heuristic method. Energies 15 (23), 9222.

Ye, C., He, W., Chen, H., 2022. Electric vehicle routing models and solution algorithms in logistics distribution: A systematic review. Environ. Sci. Pollut. Res. 29 (38), 57067–57090.

Zang, Y., Wang, M., Qi, M., 2022. A column generation tailored to electric vehicle routing problem with nonlinear battery depreciation. Comput. Oper. Res. 137, 105527.

Zhang, A., Li, T., Tu, R., Dong, C., Chen, H., Gao, J., Liu, Y., 2021a. The effect of nonlinear charging function and line change constraints on electric bus scheduling. Promet-Traffic Transp. 33 (4), 527–538.

Zhang, L., Wang, S., Qu, X., 2021b. Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile. Transp. Res. E 154, 102445.

Zhou, Y., Chen, L., Yang, Y., Li, Y., Cheng, G., Fu, Y., Zhu, C., Liu, Y., Mao, H., 2020. Electric vehicle routing problem: Model and algorithm. In: 2020 12th International Conference on Measuring Technology and Mechatronics Automation. ICMTMA, IEEE, pp. 1049–1054.

Zhou, Y., Meng, Q., Ong, G.P., 2022. Electric bus charging scheduling for a single public transport route considering nonlinear charging profile and battery degradation effect. Transp. Res. B 159, 49–75.

Zündorf, T., 2014. Electric Vehicle Routing with Realistic Recharging Models (Master's thesis). Karlsruhe Institute of Technology, Karlsruhe, Germany, Unpublished.

Zuo, X., Xiao, Y., You, M., Kaku, I., Xu, Y., 2019. A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function. J. Clean. Prod. 236, 117687.