ARTICLE IN PRESS
JID: EOR [m5G;July 15, 2022;13:55]
European Journal of Operational Research xxx (xxxx) xxx
Contents lists available at ScienceDirect
European Journal of Operational Research
journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Multi-depot electric vehicle scheduling in in-plant production logistics considering non-linear charging models

Heiko Diefenbach[a], Simon Emde[b,*], Christoph H. Glock[a]

[a] Technical University of Darmstadt, Institute of Production and Supply Chain Management, Hochschulstraße 1, Darmstadt 64289, Germany
[b] Aarhus University, CORAL - Cluster for Operations Research, Analytics, and Logistics, Department of Economics and Business Economics, Fuglesangs Allé 4, Aarhus V DK-8210, Denmark

ARTICLE INFO

Article history:
Received 1 February 2021
Accepted 24 June 2022
Available online xxx

Keywords:
Scheduling
Non-linear charging
Tow train
In-plant logistics
Electric vehicle scheduling

ABSTRACT

Electric vehicle scheduling is concerned with assigning a fleet of electrically powered vehicles to a set of timetabled trips. Since the range of these vehicles is limited, charging breaks need to be scheduled in-between trips, which require detours and time. This paper presents a novel electric vehicle scheduling problem with multiple charging stations in an in-plant logistics setting with the objective of minimizing the required fleet size. Contrary to previous works, we consider constant, linear and non-linear battery charging functions, which, among other things, allows to model realistic non-linear lithium-ion battery charging. We present an integer programming model and an exact branch-and-check solution procedure, which is based on decomposing the problem into a master and a subproblem. The former is concerned with assigning vehicles to trips while relaxing the battery constraints. The latter schedules charging breaks and checks if the master problem's solution is feasible with regard to the non-relaxed battery constraints. Our computational tests show that solving the IP model with a standard solver (CPLEX) is inferior to the branch-and-check approach, which generally performs well even for practically relevant instance sizes. Furthermore, we derive some insights into the influence of the charging mode and maximum battery capacity on the required fleet size. Lastly, we investigate the effects of the number of warehouses (with respective charging stations).

1. Introduction

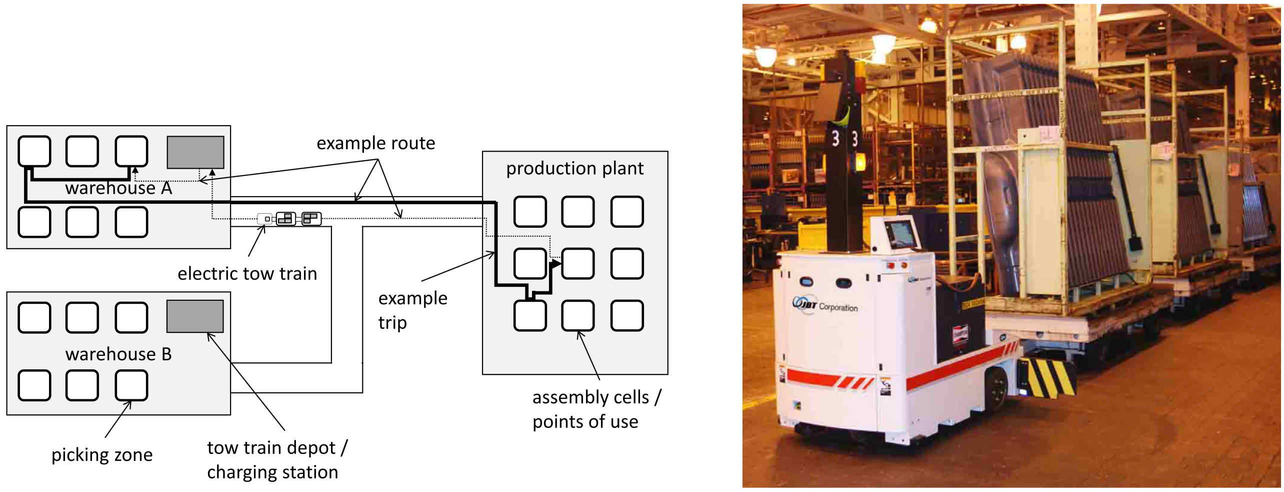
Vehicle scheduling is concerned with assigning a set of timetabled trips to a set of vehicles to satisfy some objective, typically the minimization of the size of the vehicle fleet, or of the total deadheading time, or of a combination of both. While vehicle scheduling problems have received a lot of attention from academia, interest in scheduling electric vehicles, whose range is severely limited by their battery, has only recently increased.

This paper considers the multi-depot electric vehicle scheduling problem with the objective of minimizing the fleet size (EVSP-MD-FS). It can be briefly described as follows. Given are a set of timetabled service trips with predetermined start and end times, a fleet of homogeneous electric vehicles with a limited battery capacity, a set of depots for the vehicles, and a set of charging stations (which can correspond to the depots), where the vehicles can be charged. Which vehicle should process which service trip such that no battery ever runs empty, the timetabled processing times are met, and the total fleet size is minimal? Batteries can be (partially or fully) charged in-between service trips at the charging stations, time permitting. We consider constant, linear and non-linear battery charging functions, which allows modeling the two most common charging technologies, battery swapping and lithium-ion battery plug-in charging with a realistic, non-linear charging function. In contrast to many classic versions of vehicle scheduling problems (VSP) and electric vehicle scheduling problems (EVSP), we consider deadheading trips solely in the form of additional time and battery requirements, but not as costs in the objective function.

EVSP-MD-FS is motivated by an application in in-plant production logistics, where the typical part feeding process in mixed-model assembly plants is as follows (Boysen, Emde, Hoeck, & Kauderer, 2015; Emde, Abedinnia, & Glock, 2018). Customer orders are sequenced several days before production begins, such that the exact part demand in every work cycle is known with certainty. Subsequently, service trips – trips for re-supplying the work cells –, schedules, and loads of the tow trains – electric vehicles attached to a handful of wagons carrying parts to the assembly line, see

* Corresponding author.
E-mail addresses: diefenbach@pscm.tu-darmstadt.de (H. Diefenbach), siem@econ.au.dk (S. Emde), glock@pscm.tu-darmstadt.de (C.H. Glock).

https://doi.org/10.1016/j.ejor.2022.06.050



(a) Example of a tow train making a delivery on factory premises. (b) Electric tow train[1].

Fig. 1. Electric vehicles carrying parts to final assembly.

Fig. 1b – are determined. Since space at the work cells on the shopfloor is scarce, it is usually not possible to store parts for more than a few work cycles there. Hence, deliveries are frequent and come in small lots. Logistics workers in the warehouses prepare the parts needed at the assembly line in stillages. A service trip – or trip for short – by the tow train then consists of the following steps: pick-up of the prepackaged stillages from the respective picking stations in the warehouse, drive to the assembly plant, and drop-off of the stillages at the appropriate assembly cells. After completing a trip, the tow train may set off to the start point of a consecutive trip, to a recharging station or, at the end of the shift, to a depot, where tow trains are also stationed at the beginning of the shift. The parts that need to be picked, the picking zones involved, and the assembly stations that need to be visited on a given trip depend on the production sequence. Since tow trains are electric (combustion engines are generally not legally allowed on shopfloors), the vehicles' batteries must be periodically charged at charging stations.

In classic VSP, the minimization of deadheading distances is often the primary objective. However, this is not the case for the application in in-plant production logistics due to the following properties of the latter. Apart from distances being generally shorter, routes are frequently circular milk-runs: a vehicle sets off from a logistics area, executes a service trip, and returns to the logistics area for the next trip. Therefore, the vehicles pass by depots and stations quite frequently. Because of this and due to the short driving distances, deadheading – while still necessary to some degree – is less of a concern in production logistics. Instead, the number of vehicles and operators are the main cost drivers, which, therefore, should be minimized (Emde et al., 2018; Golz, Gujjula, Günther, Rinderer, & Ziegler, 2012). It is essential to minimize the fleet size on an operational level, since operational decisions entail the satisfaction of tactical objectives – requiring fewer vehicles for daily operations enables a company to reduce their (tactical) fleet size in the long run.

Throughout the paper, we distinguish trips from routes. Trips always refer to pre-planned service trips as explained above. A vehicle's route is a sequence of consecutively executed trips, visited stations, and depots. An exemplary tow train's route, where the charging stations correspond to the depots and the executed trip is marked in bold, is depicted in Fig. 1a.

Vehicles for in-plant production logistics, such as tow trains, for example, are usually equipped with lithium-ion batteries. For these vehicles, two approaches are common to replenish empty batteries: either battery swapping (i.e., exchanging a depleted battery with a charged one) or plug-in charging. For the latter, either a constant current-constant voltage (CC-CV) scheme or a constant power-constant voltage scheme (CP-CV) is usually applied (Liu, 2013). Both charging schemes consist of two phases and aim to reduce battery degradation by avoiding excessive voltage levels during charging. In the first phase, the battery is charged with a constant current (CC-CV) or power (CP-CV), which results in a linear increase of the battery's state-of-charge (SOC) over time. At some point, usually when the SOC is at about 80%, the battery's voltage reaches a critical level and the second phase begins. Here, the voltage is kept constant at the maximum acceptable level resulting in a declining charging rate over time such that the battery's SOC gradually approaches the maximum charge. The resulting charging function is a concave piece-wise non-linear function (cf., Fig. 3a).

The contribution of this paper is as follows. We present a novel application for the multi-depot EVSP in in-plant logistics, thereby extending the electric vehicle milk-run scheduling problem, first introduced by Emde et al. (2018), by considering multiple depots. Contrary to most previous works on the EVSP, we consider realistic lithium-ion battery charging functions, including battery swapping and non-linear charging functions. We develop an IP model and propose the first exact solution method, based on branch-and-check, for this kind of problem with non-linear battery charging. The solution approach is shown to solve instances generated based on realistic specifications in good time. Moreover, we provide some insight into the effect of the charging technology as well as warehouse and depot placement on system performance.

The remainder of this paper is organized as follows. We review the literature in Section 2. In Section 3, we formally define the problem. A compact integer programming (IP) model is provided in Section 4. We propose our branch-and-check solution method in Section 5, and test it in a computational study in Section 6. Finally, Section 7 concludes the paper.

2. Literature review

Since the seminal work of Saha (1970), VSP have received a lot of attention from academia. Bunte and Kliewer (2009) and Ibarra-Rojas, Delgado, Giesen, and Muñoz (2015) provide surveys. Scheduling of electric vehicles has been addressed considerably less often. There are some models that impose a maximum length or duration on schedules for individual vehicles (Freling & Paixão, 1995; Haghani & Banihashemi, 2002; Wang & Shen, 2007); however, this is not the same as taking into consideration battery capacities and charging intervals. Only in recent years, the electric vehicle scheduling problem (EVSP) has gained increasing attention from the scientific community. The majority of research deals with the EVSP based on the application in public transportation. Although the problems discussed in the context of public transportation have a different focus than EVSP-MD-FS in terms of objectives (mostly minimization of deadheading cost), problem structure (e.g., chaining multiple visits to charging stations in a row), and instance data (e.g., the operational area is much larger, deadheading trips are substantially shorter than service trips, etc.), from a modelling perspective they are nonetheless similar. We therefore discuss the state of the art in public transport EVSP in more detail in the following.

Wen, Linde, Ropke, Mirchandani, and Larsen (2016) discuss a multi-depot electric VSP, where vehicles have a limited battery capacity and can be charged at any given charging station. They assume that batteries are charged at a linear rate (i.e., battery swapping is not considered) and that the energy consumption depends on the driving distance. Moreover, they allow for chains of charging, i.e., visiting a sequence of charging stations between trips. The goal is to minimize the weighted sum of the fleet size and the total deadheading distance. They present a heuristic based on adaptive large neighborhood search.

Wang, Huang, Xu, and Barclay (2017) present a case study of the electric mass transit system in Davis, California. They develop a holistic optimization model that considers the total annual operating costs, including, among other things, charging costs and the cost for setting up the charging stations in the first place. Similarly, Chao and Xiaohong (2013) present a case study from Chinese metropolitan areas. Messaoudi and Oulamara (2019) study the operational problem of optimizing the deployment of electric buses based on a case study at an urban public transportation service. They assume that buses can only charge at a central depot. Besides scheduling charge events, they aim to optimize the assignment of buses to line services and to parking positions with varying accessibility at the depot.

Reuer, Kliewer, and Wolbeck (2015) and Adler and Mirchandani (2017) investigate the electric VSP where vehicles must always be charged to capacity, i.e., no partial charging is allowed. The latter assume a constant time for battery charging and allow for chains of charging with the aim to minimize the fleet size and operational costs for charging. They solve the considered problem using a branch-and-price approach. Reuer et al. (2015) assume that charging times are constant and minimize the total deadheading cost. Besides only considering the scheduling of electric vehicles, Reuer et al. (2015) also consider a scheduling problem with a combined fleet of electrical and conventional vehicles, where the latter have unlimited range. They also assume that charging stations correspond to the start or end points of a subset of the trips, such that no detours are necessary to reach charging stations.

Based on the model and a heuristic solution approach developed by Adler and Mirchandani (2017), Olsen and Kliewer (2018) study the validity of the often assumed linear and constant-time battery charging compared to actual non-linear charging. They conclude that the assumption can cause significant discrepancies and suggest a more sophisticated, i.e., non-linear, modeling approach. Olsen and Kliewer (2020) follow this stream of research further, finding that constant-time charging (as an approximation of non-linear charging) overestimates the time required to charge, which results in too many vehicles being used. On the other hand, linear charging is found to underestimate charging times, which results in too tight vehicle schedules that are infeasible in practice. Moreover, Olsen and Kliewer (2020) investigate the number of vehicles charging simultaneously at the same station. They find that the maximum number of simultaneously charging vehicles ranges from about two to eight, depending on the exact instance and the assumed battery charging model.

Li (2014) develops two models, one for electric bus scheduling and one for scheduling buses with limited range that cannot be charged between tours, however. For the electric buses, either battery swapping or fast charging is assumed. Both are modeled as constant-time charging. The objective is to minimize the weighted fleet size and total operational costs. An exact branch-and-price solution approach is developed and shown to perform well on real-world instances.

Sassi and Oulamara (2017) study the scheduling of a given fleet of both electric and conventional vehicles with the objective of minimizing the total charging and traveling cost, where they only consider plug-in charging. Similarly, Zhou, Xie, Zhao, and Lu (2020) study the scheduling of a fleet of electric and conventionally powered vehicles with the aim to minimize the total operational costs as well as carbon emissions. Moreover, Yao, Liu, Lu, and Yang (2020) consider an EVSP with multiple types of electric buses, where certain trips can only be performed by certain types of buses. They further assume that vehicles can only charge at their nearest depot or base depot, where charging is modeled as constant-time charging.

Janovec and Kohání (2019a) study the effects of degrading battery capacities on an EVSP with the objective of minimizing the fleet size. They assume linear battery charging, model the problem as a MIP and solve it using commercial software. The same model and solution approach is applied to solve real-world instances in Janovec and Kohání (2019b).

While most of the previously considered works assume the trip timetables as given, Teng, Chen, and Fan (2020) investigate the integrated optimization of trip timetables and vehicle schedules with the objective of minimizing the fleet size and charging costs. They assume that vehicles can only be charged at a single depot and must always be charged to the maximum battery capacity. They solve the integrated problem heuristically using particle swarm optimization. Somewhat related, Schneider, Stenger, and Goeke (2014) and Goeke and Schneider (2015) are concerned with the routing of electric vehicles. Their aim is to route a fleet of electric vehicles such that, for a set of given locations, each location is visited within a respective time window. The electric vehicles have a limited range and must visit stations to charge. While the former consider a fleet of pure electric vehicles, the latter assume a mixed fleet of electric and conventionally powered vehicles.

In contrast to the work presented above, where battery charging is assumed to be linear or executed in constant time, van Kooten Niekerk, van den Akker, and Hoogeveen (2017) develop a set of EVSP models with more sophisticated battery charging. Their objective is to minimize the total cost, which depends on the fleet size, the total energy cost and the battery depreciation. They allow for non-linear charging and daytime-dependent energy costs when charging. Furthermore, they track the batteries' SOC over all dis-/ charging events to assess the costs of battery depreciation. The authors formulate a MIP with simplified, linear battery charging and solve it using a commercial standard solver. In addition, they formulate a model based on column generation, where the non-linear battery charging is incorporated into the models via discretization. Based on this model, they are able to solve smaller problem instances to optimality and larger instances heuristically. However, contrary to most of the previously mentioned literature and the problem considered in this paper, van Kooten Niekerk et al. (2017) assume that charging can take place only at the end/ start of certain trips, such that no detours are necessary to visit charging stations.

Zhang et al. (2021) consider an electric bus scheduling problem with multiple types of buses and multiple depots. They assume that buses can charge in-between any two service trips. However, in contrast to the majority of EVSP including EVSP-MD-FS, they assume that each vehicle can only charge at its respective home depot, which limits the flexibility of schedules significantly. Their considered objective is to minimize the total operational costs of the fleet, which includes vehicle purchasing costs, energy consumption costs and time requirement costs. Moreover, they discuss two possibilities to approximate non-linear battery charging – either by a single linear function or by a piece-wise linear function. To solve their problem, Zhang et al. (2021) present a MIP as well as a large neighborhood search heuristic. In their computational study, the authors find that the piece-wise linear approximation works well. On the other hand, the linear approximation is shown to frequently cause too tight schedules that are infeasible in practice.

Sweda, Dolinskaya, and Klabjan (2016) examine a somewhat different problem: given a single vehicle following a given route, where should the vehicle stop to recover how much charge such that the total charging cost is minimal?

To power electric vehicles, various types of batteries and battery replenishment technologies have been developed. However, in recent years, lithium-ion batteries have been established as the dominant battery technology for electric vehicles (Den Boer, Aarnink, Kleiner, & Pagenkopf, 2013). Their main advantage is a high power density compared to other battery technologies such as, for example, Pb-acid, NiCd, or NiMH batteries (Den Boer et al., 2013). On the downside, lithium-ion batteries are expensive and have a limited lifespan, which is further influenced by various handling and employment conditions (Barré et al., 2013). Among other factors, a greater depth of discharge, higher (dis-)charging rates and a higher SOC during storage can accelerate battery degradation (Vetter et al., 2005).

Considering the high costs of lithium-ion batteries and the influence of employment conditions on their lifetime, Pelletier, Jabali, Laporte, and Veneroni (2017) suggest to integrate battery degradation into decision support models for electric vehicle operations and discuss respective modeling approaches. However, according to our observations, such considerations are currently only of minor concern for intra-logistical planners and only taken into account in a rudimentary manner, for example, by limiting the depth of discharge of battery powered vehicles. The consideration of battery degradation is therefore beyond the scope of this paper.

Finally, from a strategic point of view, research has been concerned with planning optimal locations of charging stations in the context of public transportation. Liu, Song, and He (2018), for example, consider the location planning of fast-charging stations for electric bus systems. They develop both an exact and a robust stochastic optimization model with the objective of minimizing the total costs for charging stations and required batteries. He, Song, and Liu (2019) consider a similar problem, where they additionally account for costs of energy storage systems and electricity demand. Lin, Zhang, Shen, Ye, and Miao (2019) present a large-scale charging station planning approach that not only considers charging demands but also the available power grid infrastructure. The strategic planning of charging locations is, however, beyond the scope of this paper.

To summarize the previous review and to emphasize the contribution of this paper, we conclude the overview of the EVSP with Table 1, which compares the versions of EVSP that are most closely related to the problem considered in this paper. The overview demonstrates the novelty of this paper by considering realistic, non-linear battery charging in a situation where charging requires detours and vehicles are free to charge at any charging station. Moreover, most VSP papers consider the problem in the context of public transportation, which requires a different focus on things like, e.g., deadheading costs and multiple visits of charging stations in a row due to the sparse charging infrastructure relative to the large service area. To the best of our knowledge, the only previous paper dealing with electric VSP in a production logistics context is Emde et al. (2018), who present a tabu search heuristic for the single depot case. Unlike this previous paper, we consider both constant-time and non-linear charging as well as multiple depots and develop an exact algorithm.

Given that tow trains are widely employed in many production systems, it is not surprising that several publications deal with the routing and scheduling of such vehicles in a just-in-time context (e.g., Emde & Boysen, 2012; Emde & Gendreau, 2017; Emde & Schneider, 2018; Fathi, Rodríguez, & Alvarez, 2014; Fathi, Rodríguez, Fontes, & Alvarez, 2016; Zhou & Peng, 2017). Despite the fact that in-plant delivery vehicles are almost without exception electric ones, very few papers on automotive part feeding have so far considered this aspect. Hu, Zhou, and Li (2017) address the problem of drawing up a schedule for a single electric tow train serving a given route, where the speed of the vehicle can be adjusted to modify its energy consumption, which is to be minimized. Zhou and Tan (2018) optimize both the location of charging stations and the routes of the tow trains in a holistic approach. Finally, Briand, He, and Ngueveu (2018) aim to schedule tow trains such that their energy consumption is minimal. They also propose a method to derive realistic energy expenditures, showing that minimizing travel distance and minimizing energy expenditure are not always the same. All of these papers consider the superordinate problems of deciding on routes and schedules for the vehicles, not assigning individual vehicles to already timetabled trips. An overview of the whole part logistics process in the automotive industry is given by Boysen et al. (2015).

From a scheduling perspective, the EVSP-MD-FS is somewhat reminiscent of scheduling point-to-point deliveries with time windows (Emde & Zehtabian, 2019; Gschwind, Irnich, Tilk, & Emde, 2019), except that trips do not only have time windows, but that they are already timetabled. Scheduling jobs with fixed starting times on a set of machines is the subject of interval scheduling (surveyed by Kolen, Lenstra, Papadimitriou, & Spieksma, 2007; Kovalyov, Ng, & Cheng, 2007), which does not take battery charging into account, however.

3. Problem description

We present a formal description of the EVSP-MD-FS in Section 3.1 and an IP model in Section 4. Like all mathematical models, our problem definition is based on some assumptions.

Specifically, we assume that all parameters, especially all trips, are known and deterministic, which is a realistic assumption in just-in-time industries, where production sequences are typically fixed several days in advance, and the exact transport demands are consequently known (Emde, Fliedner, & Boysen, 2012). Moreover, we consider homogeneous vehicles with fully charged batteries at the beginning of the planning horizon, which are initially all stationed at depots.

All charging stations always have sufficient space and capacity to service any docked tow trains. Clearly, this may not always be the case in practice, where charging capacities can be limited by space or the charging infrastructure's capacity. Nevertheless, it presents a common assumption in the literature (e.g., Reuer et al., 2015; Wen et al., 2016; Zhang et al., 2021). Moreover, Olsen and Kliewer (2020) found that even without restricting the stations' capacities in their model, solutions rarely had an excessive number of vehicles charging simultaneously, which is in line with our observations in practice. Finally, we consider deadheading implicitly in the constraints, but not in the objective function, because vehi-

Table 1
Comparison of different electric vehicle scheduling problems.

paper	battery charging				charge tours		objective			depots		bat. per distance		other constraints	
	constant t. charging	linear t. charging	non-linear t. charging	partial charging possible	charging requires detour	free choice of charging st.	miscellaneous	w. FS + w. total TC/ RC	FS	multiple	single	proportional	arbitrary	start de. equals end de.	de. have max. ve. cap.
Chao and Xiaohong (2013)	✓						✓[8]		✓	✓	✓		✓		✓
Li (2014)[1]	✓				✓	✓	✓[9]	✓	✓		✓		✓		✓
Reuer et al. (2015)[2]	✓					✓		✓	✓		✓	✓			
Wen et al. (2016)		✓		✓	✓	✓		✓	✓	✓	✓	✓		✓	
Adler and Mirchandani (2017)	✓				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Sassi and Oulamara (2017)[2]		✓		✓		✓	✓[10]		✓	✓	✓	✓			
Wang et al. (2017)	✓			✓	✓	✓	✓[11]				✓	✓			
van Kooten Niekerk et al. (2017)	✓	✓	✓	✓		✓	✓[12]	✓	✓	✓	✓	✓	✓	✓	
Emde et al. (2018)	✓						✓[13]		✓		✓	✓	✓		
Olsen and Kliewer (2018)[3]	✓		(✓)[6]		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Janovec and Kohání (2019a)[4]		✓		✓	✓	✓			✓		✓	✓	✓		
Messaoudi and Oulamara (2019)		✓		✓				✓	✓		✓	✓	✓		
Olsen and Kliewer (2020)	✓	✓	(✓)[6]	✓	✓	✓	✓[14]	✓	✓		✓	✓	✓		
Yao et al. (2020)	✓				✓		✓[15]		✓	✓	✓	✓	✓	✓	
Zhou et al. (2020)[5]		✓		✓	✓		✓[16]	✓	✓		✓	✓			
Zhang et al. (2021)		✓	(✓)[7]	✓	✓		✓[17]	✓	✓	✓	✓	✓	✓	✓	✓
this paper	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓

Abbreviations: bat. = battery; cap. = capacity; de. = depot; FS = fleet size; max. = maximum; TC/ RC = travel cost and/ or charging cost; ve. = vehicle; w. = weighted. Annotations: [1] This refers to the considered EVSP with electric buses.; [2] This refers to the considered EVSP with electric vehicles only.; [3] Their model is identical to the one of Adler and Mirchandani (2017).; [4] The problem is identical to Janovec and Kohání (2019b).; [5] This refers to the considered electric buses within their mixed-fleet model.; [6] The solution process only considers constant-time/ linear charging – non-linear charging is only considered to examine solutions afterwards.; [7] Non-linear charging is approximated using a piece-wise linear function; [8] The objective is to minimize the cost of the fleet size, standby batteries, and total charge demand.; [9] The considered objective is to minimize the fleet size, the total travel distance, and total battery charging service cost.; [10] The objective is to maximize the distance covered with electric buses (instead of conventional ones), and minimize their energy consumption.; [11] The objective is to minimize the annual total electric bus recharging system operating costs.; [12] The considered objective is to minimize the fleet size, total energy cost, and total battery depreciation costs.; [13] The objective is to minimize the fleet size and the workload of the busiest vehicle.; [14] The objective is to minimize the fleet size plus the operational costs.; [15] The objective is to minimize the total annual scheduling costs.; [16] The objective is to minimize the operative costs and carbon emissions; [17] The objective is to minimize the operational costs including vehicle purchasing costs, energy consumption costs, and travel time costs.

Table 2
Notation.

c	state of charge (SOC)
$\hat{c}$	maximum SOC
D	set of depots; $D = \{1, \ldots, d\}$
J	set of trips; $J = \{d+1, \ldots, d+n\}$
j	depot, trip or station index
S	set of charging stations; $S = \{d+n+1, \ldots, d+n+s\}$
$\tilde{c}_j$	required charge to complete trip j
$\bar{c}_{j,j'}$	required charge for the transit from depot, station, or (the end of) trip j to depot, station, or (the start of) trip j'
e_j	end time of trip j
$\bar{m}_j$	maximum vehicle capacity of depot j
m	number of vehicles
r	constant time required to attach a vehicle to a charger (or to exchange the battery, depending on the context)
s_j	start time of trip j
$t_{j,j'}$	transit time from depot, station, or (the end of) trip j to depot, station, or (the start of) trip j'
$\delta_k(l)$	mapping signifying that vehicle k visits station $\delta_k(l) \in S$ or no station at all ($\delta_k(l) = 0$) after the lth trip of its schedule
$\zeta(j, j', j'', c)$	function giving the maximum SOC at the end of trip (or depot) j' for a vehicle coming from j with SOC c visiting station $j'' \in S$ or no station at all ($j'' = 0$) in-between
$\bar{\zeta}(k, l)$	SOC at the end of the lth trip of vehicle k
$\eta(\tau)$	battery charging function dependent on time τ; $\eta(\tau)$ is the SOC if an empty battery is charged for time τ
τ	amount of time
$\bar{\tau}(j, j', j'', c)$	time a vehicles spends charging at station j'' while transiting from j with SOC c to j'
ω_k	ordered set denoting which depots are to be visited and trips are to be performed by vehicle k in what order

cle and operator costs tend to dominate the detour costs due to the relatively short driving distances (Emde et al., 2018; Golz et al., 2012).

3.1. Formal problem description

We base the definition of the EVSP-MD-FS on the notation summarized in Table 2. Let $D = \{1, \ldots, d\}$ be the set of depots, let $J = \{d+1, \ldots, d+n\}$ be the set of trips to be processed during the planning horizon and let $S = \{d+n+1, \ldots, d+n+s\}$ be the set of charging stations (called "stations" in the following). Note that we introduce offset indices to ease notation in the following definitions and models. Generally speaking, the locations of stations do not have to coincide with the depots or the start or end points of the trips, but can of course correspond to these locations. In fact, the former is the case for most of the in-plant milk-run logistic systems we observed. Each trip consists of a vehicle visiting multiple locations (i.e., assembly cells) in a given path. The exact path the vehicle takes is immaterial for the EVSP-MD-FS; we assume that this issue has been decided in a previous step. Let s_j be the start time of trip j, i.e., the time when the vehicle arrives at the trip's first location, and let e_j be the end time of trip j, i.e., the time when the vehicle departs from the trip's last location. For notational convenience, we also define $s_j = \infty$ and $e_j = 0$, $\forall j \in D \cup S$, which implies that depots and stations are always available. Without loss of generality, we assume that $s_j \le s_{j'}$ holds, $\forall j, j' \in J : j < j'$, i.e., the trips are indexed according to their starting time in ascending order. Moreover, let $t_{j,j'}$, $\forall j, j' \in D \cup S \cup J : j \neq j'$, be the transit time of a vehicle from depot, station, or the end of trip j to depot, station, or the starting point of trip j'. For transit times, we assume that the triangle inequality holds in the sense that $t_{j,j'} \le t_{j,j''} + \max\{e_{j''} - s_{j''}, 0\} + t_{j'',j'}$, $\forall j, j', j'' \in J \cup D \cup S$, where $\max\{e_{j''} - s_{j''}, 0\}$ is the trip's execution time if $j'' \in J$, and zero if $j'' \in D \cup S$. Let the battery capacity, i.e., maximum SOC, of the vehicle be $\hat{c}$, and the charge required to complete trip $j \in J$ be $\tilde{c}_j$, where we assume $\tilde{c}_j \le \hat{c}$ applies, $\forall j \in J$. Note that the required charge $\tilde{c}_j$ may depend on the total distance covered, the number of stopovers, the carried load, the terrain (e.g., steep slopes), and other factors. We assume that these values have been determined beforehand and are given. To ease notation, we further define $\tilde{c}_j = 0$ and $\bar{c}_{j,j} = 0$, $\forall j \in D \cup S$. Let $\bar{c}_{j,j'}$ be the charge required to drive from depot, station or the end of trip j to depot, station, or the starting point of trip j'. As for travel times, we assume that the triangle inequality also holds for battery requirements in the sense that $\bar{c}_{j,j'} \le \bar{c}_{j,j''} + \tilde{c}_{j''} + \bar{c}_{j'',j'}$, $\forall j, j', j'' \in J \cup D \cup S$. We assume that charging is only possible at stations. If vehicles should be able to charge at depots or at the start or end points of trips, we can add stations that correspond to the respective locations.

Let $\eta(\tau)$ denote the charging function over time τ, i.e, the SOC an empty battery would have after being charged for a time of τ. We assume that $\eta(\tau)$ is either a constant function, a linear function or a non-linear function, which, in the latter two cases, is capped at $\hat{c}$, i.e., when the battery is full. Moreover, we assume that a vehicle arriving at a station requires a time of r before charging can start, where, depending on the assumed charging technology, r is the time required to swap the battery or to connect the battery to the charger. This enables us to account for the two most commonly encountered charging technologies in practice, namely battery swapping and charging via power cable with a realistic non-linear increase of the SOC. Furthermore, we assume the charging function is equal for all vehicles and stations and that vehicles are always charged whenever they spend sufficient time at a charging station. For a vehicle charging for a time of τ beginning from a SOC $c \ge 0$, the resulting SOC is given as $\eta\left(\eta^{-1}(c) + \tau\right)$, where $\eta^{-1}(c)$ is the inverse function of $\eta(\tau)$ and states the time τ required to charge an empty battery to SOC c. For the case of $\eta(\tau)$ being a constant function, we define $\eta^{-1}(c) = 0$. To further ease notation, we define $\eta^{-1}(c) = 0$, for $c < 0$, and $\eta^{-1}(c) = \infty$, for $c > \hat{c}$, as well as $\eta(\tau) = 0$, for $\tau < 0$. This makes it easy to accommodate non-linear charging.

A solution for EVSP-MD-FS then consists of

- a collection $\{\omega_1, \ldots, \omega_m\}$ of ordered sets ω_k, denoting which depots are to be visited and which trips are to be performed by each vehicle $k = 1, \ldots, m$ in what order,
- and m mappings $\delta_k : \{1, \ldots, |\omega_k| - 1\} \to j$, for $k = 1, \ldots, m$, where $\delta_k(l) = j$ specifies the deadheading route of vehicle k between the lth trip (or depot for $l = 1$) and the $(l+1)$th trip (or depot for $l = |\omega_k| - 1$). Two general types of deadheading routes are possible: If $j = 0$, the vehicle drives directly form trip l to $l+1$ without any charging break. Otherwise, if $j \in S$, the vehicle visits station j after the lth trip (or depot) to recharge its battery before heading for the $(l+1)$th trip (or depot).

Let $\omega_k(l)$ refer to the lth element in the ordered set. For all vehicles $k = 1, \dots, m$, it must hold that $\omega_k(1) \in D$ and $\omega_k(|\omega_k|) \in D$, since each vehicle departs from a depot initially and returns to one at the end. Furthermore, all trips need to be processed exactly once, i.e., $\bigcup_{k=1}^{m} \bigcup_{l=2}^{|\omega_k|-1} \{\omega_k(l)\} = J$ and $\omega_k(l) \neq \omega_{k'}(l'),\ \forall k, k' = 1, \dots, m; l, l' = 2, \dots, |\omega_k| - 1 : k \neq k' \vee l \neq l'$ must hold.

Note that by these definitions, vehicles are also allowed (but not required) to visit a charging station right after leaving and before returning to a depot. Moreover, our definition of δ_k imposes that vehicles visit at most one station between consecutively executed service trips. This forecloses the possibility of chains of charging, which would be especially useful if distances between service trips are large compared to the vehicles' battery reach, such that multiple charges are required to travel between consecutive service trips. Since distances are comparatively small in intralogistic transportation systems, visiting multiple charging stations in direct succession is all but unheard of, however.

We say a solution to an instance of EVSP-MD-FS is feasible if it is *feasibly executable*. I.e., for every vehicle, the battery's SOC must never be negative and the executed route must not violate the time constraints. For a formal definition, see Section 3.2.

Moreover, while somewhat uncommon in the in-plant transportation systems that motivated this paper, there are additional constraints that planners may reasonably want to consider. First, due to space constraints, each depot $j \in D$ could be required to host no more than $\bar{m}_j$ vehicles, where $\bar{m}_j = n$ if no limit exists. Second, vehicles could be required to return to their depot of initial departure at the end of their schedule. In the following, we discuss them as optional extensions of the basic EVSP-MD-FS problem. Specifically, $|\{k \in \{1, \dots, m\} : \omega_k(1) = j\}| \leq \bar{m}_j \wedge |\{k \in \{1, \dots, m\} : \omega_k(|\omega_k|) = j\}| \leq \bar{m}_j,\ \forall j \in D$, must hold to account for the limited capacity of the depots. Finally, if vehicles are required to return to their depot of initial departure, it must hold that $\omega_k(1) = \omega_k(|\omega_k|),\ \forall k \in \{1, \dots, m\}$.

The objective of EVSP-MD-FS is to find a feasible schedule that minimizes the number of vehicles m. Hence, we minimize the number of vehicles m to which trips are assigned.

3.2. Feasibly executable *solutions*

In order to determine whether a solution is *feasibly executable*, we must determine the recharged battery during charging breaks and, hence, their duration. Each vehicle can charge its battery whenever it visits a station, i.e., whenever $\delta_k(l) \in S$. We can preempt the decision on the charging breaks' duration by taking the following proposition into account.

Proposition 3.1. *Given the trips (and depots) ω_k to be executed (or visited) by vehicle k, the policy of charging the vehicle at every charging station it visits for as long as the trip sequence ω_k permits creates a feasible solution, if any exists. I.e., using this charging policy, the vehicle reaches each trip (and depot) $j \in \omega_k$ with the maximum SOC possible given ω_k.*

Proof. The proof is by an interchange argument. Assume that in some feasible solution, vehicle k reaches trip $j \in \omega_k : j \neq \omega_k(|\omega_k|)$ with SOC c' and the consecutive trip (or depot) j' with c'', where it executes the deadheading route j'' in-between. We refer to this as case one. Alternatively, assume a second case, where j is reached with SOC $c''' = c' + \Delta c$ and j' with SOC c'''', where Δc is a positive amount of charge. In the following, we prove that $c'''' \geq c''$ is guaranteed to hold true.

In the second case, the vehicle can execute the exact same deadheading route as in the first case. If $j'' = 0$, it follows that $c'' = c' - \bar{c}_{j,j'}$ and $c'''' = c''' - \bar{c}_{j,j'} = c' + \Delta c - \bar{c}_{j,j'}$, hence $c'''' > c''$ applies. Else, if $j'' \in S$, in case one, the vehicle reaches j'' with SOC $c' - \bar{c}_{j,j''}$. Let τ' denote the time the vehicle charges at station j'', such that it has SOC $\eta\left(\eta^{-1}\left(c' - \bar{c}_{j,j''}\right) + \tau'\right)$ when leaving j''. Consequently, $c'' = \eta\left(\eta^{-1}\left(c' - \bar{c}_{j,j''}\right) + \tau'\right) - \bar{c}_{j'',j'}$ follows. In the second case, the vehicle reaches station j'' with $c' + \Delta c - \bar{c}_{j,j''}$ and can charge for the exact same duration τ' as in the first case. Consequently, it follows that $c'''' = \eta\left(\eta^{-1}\left(c' + \Delta c - \bar{c}_{j,j''}\right) + \tau'\right) - \bar{c}_{j'',j'}$. Since $\eta^{-1}(c)$ is an increasing function of c and $\eta(\tau)$ is an increasing function of τ, $c'''' \geq c''$ is guaranteed to hold true.

The same reasoning applies for all consecutive trips in ω_k until the final depot $\omega_k(|\omega_k|)$ is reached. Hence, if there exists a feasible solution, there also exists a feasible solution where the remaining charge when reaching each $j \in \omega_k$ is maximized, that is, where charging breaks are never cut short before the vehicle must depart for its next trip. □

Therefore, at each visited station, the vehicle should charge for the maximum possible duration that still allows to timely reach the consecutive trip or depot. Given $\delta_k(l) = j''$ with $j'' \in S$, let $\omega_k(l) = j$ and $\omega_k(l+1) = j'$ be two consecutively executed trips or one of them be a depot. Furthermore, let c be the SOC when the vehicle leaves from trip (or depot) j. The optimal time $\bar{\tau}(j, j', j'', c)$ the vehicle spends charging at station j'' then follows as

$$\bar{\tau}\left(j, j', j'', c\right) = s_{j'} - e_j - t_{j,j''} - t_{j'',j'} - r, \tag{1}$$

which is comprised of the available time between the start of trip j' and the end of trip j minus the travel time to and from station j'' minus the plug-in/ battery swap time r.

Given two consecutively executed trips $\omega_k(l) = j$ and $\omega_k(l+1) = j'$ (where either might be a depot instead) and $\delta_k(l) = j''$ (with $j'' \in S \cup \{0\}$) indicating the deadheading route in-between, the maximal SOC at the end of trip (or depot) j' can be calculated as

$$\zeta\left(j, j', j'', c\right) = \begin{cases} c - \bar{c}_{j,j'} - \tilde{c}_{j'} & \text{if } j'' = 0 \wedge e_j + t_{j,j'} \leq s_{j'} \\ \eta\left(\eta^{-1}\left(c - \bar{c}_{j,j''}\right) + \bar{\tau}\left(j, j', j'', c\right)\right) & \text{if } j'' \in S \wedge \bar{c}_{j,j''} \leq c \\ \quad -\bar{c}_{j'',j'} - \tilde{c}_{j'} & \quad \wedge \bar{\tau}\left(j, j', j'', c\right) \geq 0 \\ -1 & \text{otherwise} \end{cases}. \tag{2}$$

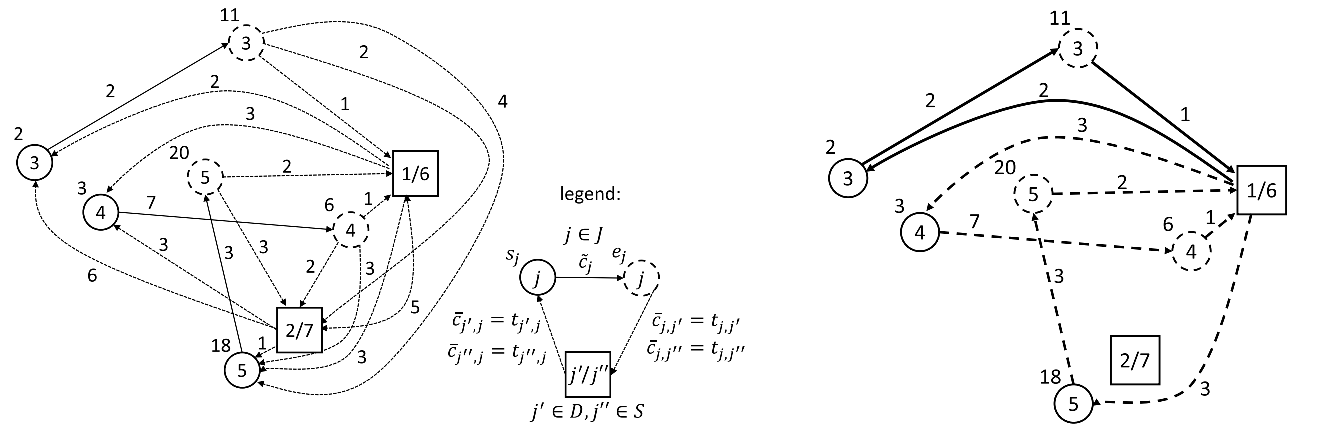
If no station is visited at the deadheading route, the first case applies. Here, the SOC at the end of trip j' is the SOC at the end of trip j minus the battery charge consumed for the transit from j to j' and for the execution of j'. If a station is visited in-between j and j', the second case applies. Here, the vehicle reaches station j'' with SOC $c - \bar{c}_{j,j''}$ and charges for a duration of $\bar{\tau}\left(j, j', j'', c\right)$, which results in a SOC of $\eta\left(\eta^{-1}\left(c - \bar{c}_{j,j''}\right) + \bar{\tau}\left(j, j', j'', c\right)\right)$ at the end of the charging procedure. Afterwards, trip j' must be reached and executed, which reduces the SOC at the end of trip j' by $\bar{c}_{j'',j'} + \tilde{c}_{j'}$. If either the charge c is not sufficient to reach the station j'' or there is not enough time, $\zeta\left(j, j', j'', c\right)$ assumes a value < 0 to indicate infeasibility.

We define $\bar{\zeta}(k, l)$ as the SOC of vehicle k at the end of the lth trip or depot, where $\bar{\zeta}(k, |\omega_k|)$ denotes the SOC when vehicle k reaches the final depot. For $l \geq 2$, it can be recursively calculated as

$$\bar{\zeta}(k, l) = \zeta\left(\omega_k(l-1), \omega_k(l), \delta_k(l-1), \bar{\zeta}(k, l-1)\right),$$

with $\bar{\zeta}(k, 1) = \hat{c}$ for $l = 1$.

Based on these definitions, a solution to EVSP-MD-FS is *feasibly executable* if and only if $\zeta(k, l) \geq 0,\ \forall k \in \{1, \dots, m\}, l \in \{1, \dots, |\omega_k|\}$. I.e., for every vehicle, the battery's SOC must never be negative and the executed route must not violate the time constraints (since, otherwise, $\zeta\left(j, j', j'', c\right)$ and, hence, $\bar{\zeta}(k, l)$, would assume the value -1).



(a) Example problem data. (b) Optimal solution in the example.

Fig. 2. Example data and solution.

Example. Consider the example problem depicted in Fig. 2a, consisting of $n = 3$ trips and $d = s = 2$ depots that are also charging stations. Let the battery capacity of the vehicles be $\hat{c} = 11$, the charge function be linear $\eta(\tau) = \tau$ (with the cap $\eta(\tau) = 11$, for $\tau \geq 11$) and the constant charge time be $r = 0$. Furthermore, let $\bar{m}_j = n$, $\forall j \in D$, and let the vehicles be not required to return to their initial depot. Note that in this example, we assume that all driving times ($t_{j,j'}$) are equal to the energy expenditures ($\bar{c}_{j,j'}$), which generally does not need to be the case. The optimal solution is depicted in Fig. 2b, corresponding to $m = 2$ vehicles being used. Solid arrows denote the route of vehicle 1, dotted arrows the route of vehicle 2. Vehicle 1 processes only trip 3 (i.e., $\omega_1 = \{1, 3, 1\}$, $\delta_1(1) = 0$, $\delta_1(2) = 0$, $\bar{\zeta}(1, 1) = 11$, $\bar{\zeta}(1, 2) = 7$, $\bar{\zeta}(1, 3) = 6$). Vehicle 2 processes trips 4 and 5 (in that order), returning to, first, station 6 – where it charges its battery for a time of $\bar{\tau}(4, 5, 6, 1) = 18 - 6 - 1 - 3 - 0 = 8$ – and, second, depot 1 after the trips (i.e., $\omega_2 = \{1, 4, 5, 1\}$, $\delta_2(1) = 0$, $\delta_2(2) = 6$, $\delta_2(3) = 0$, $\bar{\zeta}(2, 1) = 11$, $\bar{\zeta}(2, 2) = 1$, $\bar{\zeta}(2, 3) = 2$, $\bar{\zeta}(2, 4) = 0$).

3.3. Complexity

Concerning the time complexity of EVSP-MD-FS, classic (single-depot) non-electric vehicle scheduling is well-known to be tractable (Saha, 1970), as it can be reduced to finding a minimum cost perfect matching in a bipartite graph (Bertossi, Carraresi, & Gallo, 1987). This result also holds for the special single-depot case of EVSP-MD-FS where the battery capacity is not a limiting factor ($\hat{c} = \infty$). In this case, charging breaks need not be considered and the driving time t to/from the depot is immaterial. The problem then is to assign the given trips to as few vehicles as possible such that no trips overlap, which is equivalent to classic vehicle scheduling with the objective of minimizing the fleet size.

However, as soon as the battery capacity becomes a limiting factor, single-depot electric vehicle scheduling with finite battery capacity and the objective of minimizing the fleet size becomes intractable, as is proven by Emde et al. (2018, Proposition 3) by a reduction from bin packing. Obviously, this result extends to the multi-depot case as considered in EVSP-MD-FS. Finally, the result also holds if additional constraints are considered, since they merely generalize the basic EVSP-MD-FS problem.

3.4. Feasible deadheading routes

By the definitions in Section 3.1, a solution to EVSP-MD-FS consists of two distinctive parts. The ordered subsets $\omega_1, \ldots, \omega_m$ denote which depots are to be visited and which trips to be executed by which vehicle in which sequence. The mappings $\delta_1, \ldots, \delta_m$ indicate the deadheading route in-between the trips and depots for all vehicles $1, \ldots, m$. Given only ω_k, the respective feasible deadheading routes δ_k can be determined efficiently as follows.

Let $\omega_k(l) = j$ and $\omega_k(l+1) = j'$ be two consecutively executed trips or either be a depot. Furthermore, let c be the SOC when the vehicle leaves from trip (or depot) j. By Proposition 3.1, provided that any feasible solution exists at all for a given ω_k, there is always a feasible solution that reaches j' with the highest possible SOC and, consequently, completes j' with the maximum possible SOC (since $\bar{c}_{j'}$ is constant). The latter SOC is given by $\zeta(j, j', j'', c)$ (cf., Eq. (2)), which beyond j, j' and c is solely dependent on $\delta_k(l) = j''$, the deadheading route between j and j'. Hence, it follows that

$$\delta^*_k(l) = \underset{j'' \in S \cup \{0\}}{\arg\max} \left\{ \zeta\left(j, j', j'', c\right) \right\}$$

is guaranteed to be feasible (if feasibility is attainable at all). Consequently, the maximum and therefore feasible charge at the end of trip j' is given as

$$\zeta^*\left(j, j', c\right) = \max_{j'' \in S \cup \{0\}} \left\{ \zeta\left(j, j', j'', c\right) \right\}.$$

4. IP model

To enable the use of default solvers, we propose an IP model based on the notation given in Table 3. We incorporate non-linear battery charging by discretizing the SOC. I.e., the SOC can only assume discrete values $c \in C$. We note that the discretization is only required for the IP model; the solution procedure presented in Section 5 does not rely on discretization. For linear and constant-time charging, the discrete SOC is exact, as long as we assume that all problem parameters, i.e., travel times and required battery charges, are integer. For non-linear charging, using discrete SOC is an approximation. By scaling the problem parameters accordingly, we can, however, approximate the actual SOC to any desired precision.

We introduce binary variables $\beta_{j,c}$ to track the discrete SOC $c \in C$ at the end e_j of each trip $j \in J$ and when a vehicle reaches the final depot $j \in D$. Note that the number of variables $\beta_{j,c}$ grows pseudo-polynomially with $O(\hat{c} \cdot n)$. Moreover, we define set $N = \{(j, c) \in (J \cup D) \times C \mid c \neq \lfloor \zeta^*\left(j', j, c'\right) \rfloor, \forall j' \in J \cup D, c' \in C\}$ as the set of pairs (j, c) indicating that trip j is never completed with SOC c in the optimal solution. Note that set N can be determined in $O((n + d)^2 \cdot s \cdot \hat{c})$, by simply finding all combinations $j, j' \in J \cup D$: $j < j'$, $c \in C$, where $\max_{j'' \in S \cup \{0\}} \left\{ \zeta\left(j, j', j'', c\right) \right\} = \zeta^*\left(j, j', c\right) < 0$.

Table 3
Notation for the IP model.

C	set of discrete SOC values; index c; $C = \{0, \ldots, \hat{c}\}$; a value of c' refers to $\frac{c'}{\hat{c}}$ of the maximum SOC
$\upsilon_{j,j'}$	binary variable: 1, if trip j is associated with depot j', else 0
$x_{j,j'}$	binary variable: 1, if a vehicle transits from trip or depot j to trip or depot j' (with possible detours for charging in-between), else 0
$\beta_{j,c}$	binary variable: 1, if trip j is completed (or depot j is reached) with SOC c

Using these definitions, Eqs. (3) to (13) define an IP model for the case that vehicles are not required to return to their initial depot at the end of their schedules.

$$\text{[EVSP-MD-FS] Minimize } F(\boldsymbol{x}, \boldsymbol{\beta}) = \sum_{j \in D} \sum_{j' \in J} x_{j,j'} \tag{3}$$

subject to

$$\sum_{\substack{j' \in J \cup D: \\ j' > j \vee j' \in D}} x_{j,j'} = 1 \qquad \forall j \in J \tag{4}$$

$$\sum_{\substack{j \in J \cup D: \\ j < j'}} x_{j,j'} = 1 \qquad \forall j' \in J \tag{5}$$

$$\sum_{\substack{c \in C: \\ (j,c) \notin N}} \beta_{j,c} = 1 \qquad \forall j \in J \tag{6}$$

$$0 \geq x_{j,j'} + \beta_{j,c} - 1 \qquad \forall j \in J, j' \in J \cup D : j < j' \vee j' \in D, \; c \in C : (j,c) \notin N \wedge \lfloor \zeta^*(j, j', c) \rfloor < 0 \tag{7}$$

$$\beta_{j', \lfloor \zeta^*(j,j',c) \rfloor} \geq x_{j,j'} + \beta_{j,c} - 1 \qquad \forall j \in J, j' \in J \cup D : j < j' \vee j' \in D, \; c \in C : (j,c) \notin N \wedge \lfloor \zeta^*(j, j', c) \rfloor \geq 0 \tag{8}$$

$$\beta_{j', \lfloor \zeta^*(j,j',\hat{c}) \rfloor} \geq x_{j,j'} \qquad \forall j \in D, j' \in J : \lfloor \zeta^*(j, j', \hat{c}) \rfloor \geq 0 \tag{9}$$

$$\sum_{j \in J} x_{j',j} \leq \bar{m}_{j'} \qquad \forall j' \in D \tag{10}$$

$$\sum_{j \in J'} x_{j,j'} \leq \bar{m}_{j'} \qquad \forall j' \in D \tag{11}$$

$$x_{j,j'} \in \{0, 1\} \qquad \forall j, j' \in J \cup D : (j, j' \in J \wedge j < j') \vee (j \in J \wedge j' \in D) \vee (j \in D \wedge j' \in J \wedge \lfloor \zeta^*(j, j', \hat{c}) \rfloor \geq 0) \tag{12}$$

$$\beta_{j,c} \in \{0, 1\} \qquad \forall j \in J \cup D, c \in C : (j,c) \notin N \tag{13}$$

Objective function (3) minimizes the number of required vehicles. Constraints (4) and (5) enforce that every trip has exactly one preceding and one succeeding trip or depot. Constraints (6) make sure that each trip is finished with exactly one non-negative SOC.

Constraints (7) to (9) ensure consistent SOC values. Inequalities (7) prohibit the execution of trip j' after trip j has been completed with SOC c if the SOC would be insufficient for completing the trip or time constraints would be violated (i.e., $\lfloor \zeta^*(j, j', c) \rfloor < 0$). Likewise, if the SOC is sufficient and no time constraints are violated (i.e., $\lfloor \zeta^*(j, j', c) \rfloor \geq 0$), Inequalities (8) set the SOC at the end of trip j' to the appropriate value. Constraints (9) determine the SOC of the trips that are initially executed after the vehicles leave the depots.

Constraints (10) and (11) enforce limits on the number of vehicles per depot. Finally, Constraints (12) and (13) define the domains of the decision variables. Note that – in order to reduce the search space – we omit $\beta_{j,c}$ for $(j,c) \in N$, since the states can never be feasibly reached.

Moreover, to formulate the IP model, we note that determining $\zeta^*(j, j', c)$ requires a pseudo-polynomial amount $O((n+d)^2 \cdot s \cdot \hat{c})$ of pre-calculations. Furthermore, the optimal deadheading routes need to be determined from the IP's solution in a $O((n+d) \cdot s)$ post-processing procedure (cf., Section 3.4).

To enforce the optional constraint that each vehicle return to the depot of its initial departure, we introduce the binary variables $\upsilon_{j,j'}$ that are 1 if trip j is associated with depot j', and 0 otherwise. Using these variables, we extend the IP with the following equations:

$$\sum_{j' \in D} \upsilon_{j,j'} = 1 \qquad \forall j \in J \tag{14}$$

$$\upsilon_{j',j''} \geq x_{j,j'} + \upsilon_{j,j''} - 1 \qquad \forall j, j' \in J : j < j', j'' \in D \tag{15}$$

$$\upsilon_{j,j'} \geq x_{j',j} \qquad \forall j \in J, j' \in D \tag{16}$$

$$\upsilon_{j,j'} \geq x_{j,j'} \qquad \forall j \in J, j' \in D \tag{17}$$

$$\upsilon_{j,j'} \in \{0, 1\} \qquad \forall j \in J, j' \in D \tag{18}$$

Constraints (14) make sure that every trip $j \in J$ is associated with exactly one depot. Inequalities (15) force each pair of consecutively executed trips, j and j', to be associated with the same depot. Constraints (16) force the trip executed directly after leaving a depot to be associated with the respective depot. Likewise, Constraints (17) associate each final trip before returning to a depot with the respective depot. In conjunction, these constraints enforce each tour to start from and end at the same depot.

5. Branch-and-check

Our computational tests reveal that a default solver using the IP model from Section 4 performs subpar for instances with $n \geq 100$ (see Section 6.2). We therefore propose a decomposition scheme based on branch-and-check (BCH, Beck, 2010; Thorsteinsson, 2001).

The general idea of BCH is to split a complicated problem into two parts, which are individually easier to solve: a mixed-integer programming master model, which is a relaxation of the original problem, and a subproblem, which encodes the remaining variables and/or constraints, albeit not necessarily in the form of a MIP. The master model is solved using classic branch-and-bound methods. Periodically, when it is advantageous, the subproblem is solved at a node of the branch-and-bound tree to check the feasibility of the current master solution and/ or determine its exact objective value. This information is then communicated to the overarching branch-and-bound process, usually via cuts. The search terminates when there are no more unfathomed nodes in the enumeration tree left to explore. The best found solution is optimal.

For the EVSP-MD-FS, the flow is as follows. The master MIP model is concerned with assigning trips to vehicles, while relaxing the limited battery capacity. The feasibility and charging detours are determined by solving the subproblem. The master model is solved by a black-box default solver (CPLEX). Whenever the solver finds an integer solution during its branch-and-bound process, the solution is passed to the subproblem, which is solved by a polynomial-time algorithm and – in the case of limited depot capacities – by solving a (small) additional MIP. Information about the feasibility is injected into the branch-and-bound tree by way of combinatorial cuts, similar in form to logic-based (Hooker, 2011) or combinatorial (Codato & Fischetti, 2006) Benders cuts.

The master model is described in more detail in Section 5.1. The subproblem and its solution are presented in Section 5.2. Finally, the cuts derived from the subproblem are explained in Section 5.3.

5.1. Master problem

The purpose of the master model is assigning trips to vehicles. Charging breaks (and deadheading routes) are not scheduled at this stage. As in model [EVSP-MD-FS], we use variables $x_{j,j'}$ to denote whether two jobs, j and j', are processed consecutively by the same vehicle. Moreover $x_{0,j}$ and $x_{j,0}$ denote that a vehicle departs from any depot or returns to any depot, respectively. The exact depots are determined later in the subproblem. Unlike before, we omit the binary variables $\beta_{j,c}$ and use continuous variables $\tilde{\beta}_j$, $\forall j \in J$, to calculate upper and lower bounds on the remaining SOC at the end of each trip.

Some trips can clearly not be assigned to the same vehicle due to either overlapping execution times or insufficient battery capacity. Therefore, we define the set of incompatible pairs of trips as

$$I = \left\{ (j, j') \in J \times J \mid \zeta^*\left(j, j', \max\left\{\tilde{C}_j\right\}\right) < \min\left\{\tilde{C}_{j'}\right\} \right\}, \tag{19}$$

where $\tilde{C}_j = \left\{ z \in \mathbb{R} \mid \min_{j' \in D \cup S} \left\{ \bar{c}_{j,j'} \right\} \le z \le \hat{c} - \tilde{c}_j - \min_{j' \in D \cup S} \left\{ \bar{c}_{j',j} \right\} \right\}$ is the set of feasible SOC values at the end of trip j, which takes into consideration that a vehicle must be able to feasibly reach and leave j. Note that it is possible that the vehicle executes another trip after trip j' before returning to a station or depot. However, the minimum required charge is guaranteed to be at least $\min_{j'' \in D \cup S} \{ \bar{c}_{j',j''} \}$ due to the triangle inequality holding true. In the case of insufficient time, function ζ^* will always assume -1. Otherwise, the maximum SOC that can remain at the end of trip j is $\max\left\{\tilde{C}_j\right\}$. Hence, the maximum SOC at the end of trip j' is $\zeta^*\left(j, j', \max\left\{\tilde{C}_j\right\}\right)$, which must be at least $\min\left\{\tilde{C}_{j'}\right\}$ for the vehicle to feasibly continue its tour.

The master model is then given as follows.

$$\text{[MP] Minimize } \sum_{j \in J} x_{0,j} \tag{20}$$

subject to

$$\sum_{\substack{j' \in J \cup \{0\}: \\ (j,j') \notin I}} x_{j,j'} = 1 \qquad \forall j \in J \tag{21}$$

$$\sum_{\substack{j \in J \cup \{0\}: \\ (j,j') \notin I}} x_{j,j'} = 1 \qquad \forall j' \in J \tag{22}$$

$$\tilde{\beta}_{j'} \le \tilde{\beta}_j + \bar{c}^{\max}_{j,j'} + M \cdot \left(1 - x_{j,j'}\right) \qquad \forall \in j, j' \in J : (j, j') \notin I \wedge \min\left\{\tilde{C}_j\right\} + \bar{c}^{\max}_{j,j'} < \hat{c}^{\max}_{j,j'} \tag{23}$$

$$\tilde{\beta}_{j'} \le \hat{c}^{\max}_{j,j'} + M \cdot \left(1 - x_{j,j'}\right) \qquad \forall j \in J \cup \{0\}, j' \in J : (j, j') \notin I \tag{24}$$

$$\tilde{\beta}_j \ge \sum_{\substack{j' \in J \cup \{0\}: \\ (j,j') \notin I \wedge \min\left(\tilde{C}_j\right) = \bar{c}^{\min}_{j,j'}}} \bar{c}^{\min}_{j,j'} \cdot x_{j,j'} \qquad \forall j \in J \tag{25}$$

$$x_{j,j'} \in \{0, 1\} \qquad \forall j, j' \in J \cup \{0\} : (j, j') \notin \{0\}^2 \cup I \tag{26}$$

$$\tilde{\beta}_j \in \tilde{C}_j \qquad \forall j \in J \tag{27}$$

Objective function (20) minimizes the number of vehicles in use, i.e., the vehicles that initially depart from the depots. Constraints (21) and (22) enforce that each trip $j \in J$ has a predecessor and a successor, which can also be depots.

Constraints (23) and (24) calculate upper bounds on the SOC after trip j, $\forall j \in J$. For Constraints (23), we define the parameter $\bar{c}^{\max}_{j,j'}$ as the maximum increase in SOC between the end of trip j and the end of trip j'. If more charge is consumed than charged, $\bar{c}^{\max}_{j,j'}$ is negative. Formally, we define $\bar{c}^{\max}_{j,j'} = \max_{c \in \tilde{C}_j} \left\{ \zeta^*(j, j', c) - c \right\}$. How to calculate $\bar{c}^{\max}_{j,j'}$ is explained in Appendix A.1. By Constraints (23), an upper bound on the remaining SOC $\tilde{\beta}_{j'}$ after trip j' is $\bar{c}^{\max}_{j,j'}$ plus the upper bound of the SOC $\tilde{\beta}_j$ after the preceding trip j.

For Constraints (24), we define the parameter $\hat{c}^{\max}_{j,j'}$ as the maximum SOC at the end of trip j' if j and j' are processed consecutively. Formally, we define $\hat{c}^{\max}_{j,j'} = \zeta^*\left(j, j', \max\left\{\tilde{C}_j\right\}\right)$ and $\hat{c}^{\max}_{0,j'} = \max_{j \in D} \left\{ \hat{c}^{\max}_{j,j'} \right\}$. Constraints (24) provide an additional upper bound by enforcing $\hat{c}^{\max}_{j,j'}$ as the maximum remaining charge at the end of trip j' if it is executed after trip j. Furthermore, if $\min\left\{\tilde{C}_j\right\} + \bar{c}^{\max}_{j,j'} \ge \hat{c}^{\max}_{j,j'}$ applies, the upper bound from Constraints (23) is guaranteed to be weaker than the respective one from Constraints (24), which is why we omit these in Constraints (23). Additionally, Constraints (24) provide upper bounds for the trips executed directly after the vehicles leave the initial depots. Finally, note that if two trips j and j' are not executed consecutively, $M \cdot \left(1 - x_{j,j'}\right)$ makes sure that no bound is forced upon $\tilde{\beta}_{j'}$. Since $0 \le \tilde{\beta}_{j'} \le \hat{c}$ and $\hat{c}^{\max}_{j,j'} \ge \bar{c}^{\max}_{j,j'} \ge -\hat{c}$, it is sufficient to set $M = 2 \cdot \hat{c}$.

To formulate lower bounds on the SOC, we define $\bar{c}^{\min}_{j,j'}$ as the charge that must remain at the end of trip j if trip j' is processed consecutively. Formally, we define $\bar{c}^{\min}_{j,j'} = \min\left\{ c \in \tilde{C}_j \mid \zeta^*(j, j', c) \ge \min\left\{\tilde{C}_{j'}\right\} \right\}$ and $\bar{c}^{\min}_{j,0} = \min_{j' \in D} \left\{ \bar{c}^{\min}_{j,j'} \right\}$. How to determine $\bar{c}^{\min}_{j,j'}$ is explained in Appendix A.1. Note that the SOC must not only be sufficient to reach and process trip j', but also to reach a station or depot after that trip (cf., Eq. (19)). Using this definition, Constraints (25) set a lower bound on the SOC at the end of trip j, $\forall j \in J$. Moreover, we can omit the pair j and j' in the sum, if $\min\left\{\tilde{C}_j\right\} = \bar{c}^{\min}_{j,j'}$ holds, because Constraints (27) already sets an equally strict bound on $\tilde{\beta}_j$.

Finally, Constraints (26) and (27) define the domains of the decision variables.

We solve model [MP] via a commercial black-box solver (CPLEX). Given a feasible integer candidate solution $\bar{\boldsymbol{x}}$, we derive the corresponding assignment of trips to vehicles with the following iterative procedure. We define $\hat{\omega}_k$ as the correspondent to ω_k (cf., Section 3.1) without scheduled depots at the beginning and the end, i.e., $\hat{\omega}_k = \omega_k \setminus D$. To distinguish $\hat{\omega}_k$ from ω_k, we refer to the former as a vehicle's working schedule in the following. Starting with $k = u = 1$, we calculate $\tilde{\omega}^u_k = \left\{ j \in J \mid x_{\max\{\tilde{\omega}^{u-1}_k\},j} = 1 \right\} \cup \tilde{\omega}^{u-1}_k$, where $\tilde{\omega}^0_k = \left\{ \min_{j \in J \setminus \bigcap_{k'=1}^{k-1} \{\hat{\omega}_{k'}\}} \{j\} \right\}$. As long as $\tilde{\omega}^u_k \setminus \tilde{\omega}^{u-1}_k \ne \emptyset$, we increment u and repeat the calculation. Otherwise, we set $\hat{\omega}_k =$

$\tilde{\omega}_k^u$, increment k, and reset $u = 1$ to start the procedure all over. If we reach $\tilde{\omega}_k^0 = \left\{\min_{j\in J\setminus\bigcap_{k'=1}^{k-1}\{\tilde{\omega}_{k'}\}}\{j\}\right\} = \emptyset$, we terminate the procedure and set $m = k - 1$ for the fleet size.

The resulting assignments, i.e. $\hat{\omega}_k$, $\forall k = \{1,\dots,m\}$, are, however, incomplete, because the actual depots to be visited at the beginning and in the end still need to be determined. Moreover, they may not allow a feasible solution for the original problem, because no charging breaks are scheduled and the battery of one or more vehicles may thus be insufficient. Consequently, we separate cuts and add them to model [MP] iteratively to converge on feasible and optimal solutions.

5.2. Subproblem

Given an integer candidate solution $\bar{\boldsymbol{x}}$ that is feasible for model [MP] and the assignments $\hat{\omega}_k$, $\forall k = 1,\dots,m$, derived from it, we aim to answer two questions: First, which vehicle should depart from and return to which depot, and, second, when should which vehicle visit what station to charge, i.e., determine δ_k, $\forall k \in \{1,\dots,m\}$?

The first question, i.e., determining the start and end depot for each working schedule $\hat{\omega}_k$, can be approached by simply trying all possibilities, where the start and end depot can be enforced to be identical if required. Let Ω_k be the set of all schedules ω_k for a given working schedule $\hat{\omega}_k$, where $\Omega_k(i)$ denotes the ith schedule in Ω_k, corresponding to one specific pair of start $(\Omega_k(i)(1))$ and end $(\Omega_k(i)(|\Omega_k(i)|))$ depot. If the depots' capacities are unlimited, for a given working schedule $\hat{\omega}_k$, it is sufficient to find a single feasible start and a single feasible end depot, i.e., one feasible schedule $\omega_k \in \Omega_k$. We describe how to determine if a certain ω_k is feasible later in the section.

However, if the depots' capacities are limited, we must ensure that they are not exceeded, which we do as follows. Let $\hat{\Omega}_k \subseteq \Omega_k$ be the set of feasible schedules ω_k for a given working schedule $\hat{\omega}_k$. Let $y_{i,k}$ be binary variables that are one if the ith schedule in set $\hat{\Omega}_k$ is selected and zero otherwise. I.e., if $y_{i,k}$ assumes one, working schedule $\hat{\omega}_k$ is matched with depots $\hat{\Omega}_k(i)(1)$ and $\hat{\Omega}_k(i)\left(\left|\hat{\Omega}_k(i)\right|\right)$ as initial and final depot to form a proper schedule ω_k. Moreover, let z_k be binary variables that are one if no schedule in $\hat{\Omega}_k$ is selected and zero otherwise. I.e., if z_k assumes one, working schedule $\hat{\omega}_k$ has not been matched with an initial and final depot. Based on these definitions, we formulate the following integer program (IP):

$$\text{[Depot-Matching] Minimize } Z = \sum_{k\in\{1,\dots,m\}} z_k \tag{28}$$

subject to

$$\sum_{i\in\left\{1,\dots,|\hat{\Omega}_k|\right\}} y_{i,k} + z_k = 1 \qquad \forall k \in \{1,\dots,m\} \tag{29}$$

$$\sum_{k\in\{1,\dots,m\}} \sum_{\substack{i\in\left\{1,\dots,|\hat{\Omega}_k|\right\}:\\ \hat{\Omega}_k(i)(1)=j}} y_{i,k} \le \bar{m}_j \qquad \forall j \in D \tag{30}$$

$$\sum_{k\in\{1,\dots,m\}} \sum_{\substack{i\in\left\{1,\dots,|\hat{\Omega}_k|\right\}:\\ \hat{\Omega}_k(i)(|\hat{\Omega}_k(i)|)=j}} y_{i,k} \le \bar{m}_j \qquad \forall j \in D \tag{31}$$

$$y_{i,k} \in \{0,1\} \qquad \forall k \in \{1,\dots,m\}, i \in \left\{1,\dots,|\hat{\Omega}_k|\right\} \tag{32}$$

$$z_k \in \{0,1\} \qquad \forall k \in \{1,\dots,m\} \tag{33}$$

Objective (28) minimizes the number of vehicles where no feasible schedule could be selected, i.e., where no three-dimensional matching between a working schedule, the initial, and the final depot was found. If objective Z is zero, the solution is feasible for the original problem, i.e., there exists a combination of feasible schedules such that no depots' capacity is violated. Else, we need to add cuts to [MP], as we explain in Section 5.3.

Constraints (29) enforce that either exactly one or no schedule ω_k is selected for every vehicle k. Constraints (30) and (31) ensure the depots' capacities are not exceeded by the departing and returning vehicles, respectively. Finally, Constraints (32) and (33) define the domains of the decision variables. We solve [Depot-Matching] using a default solver.

Generally, [Depot-Matching] is NP-hard since it presents a three-dimensional matching problem. Nevertheless, it may still be solved quickly by a standard solver, due to the number of variables being in $O(m\cdot s^2)$ and usually not exceeding a couple of thousands in practical application. Moreover, if vehicles are required to return to their initial depots at the end of their schedules, [Depot-Matching] reduces to a bipartite matching problem, which is known to be solvable in polynomial time.

Concerning the determination of charging breaks and detours, let ω_k be the ith element in Ω_k. The optimal deadheading routes $\delta_k(l)$, $\forall l = 1,\dots,|\omega_k| - 1$, can be derived in polynomial time by determining $\zeta^*(j, j', c)$ for every two consecutive elements, j and j', in ω_k and tracking the SOC c accordingly. More formally, we apply the following $|\omega_k|$-step iterative procedure. Starting from the initial state $l = 1$, for each state $l = 2,\dots,|\omega_k|$, we calculate $\bar{\zeta}(k,l) = \zeta^*\big(\omega_k(l-1), \omega_k(l), \bar{\zeta}(k,l-1)\big)$, where $\bar{\zeta}(k,1) = \hat{c}$ is the initial SOC. If we reach a state l, where $\bar{\zeta}(k,l) \le \min\left\{\tilde{c}_{\omega_k(l)}\right\}$, the assignment ω_k cannot be solved in a feasible manner. In this case, we save the latest state $l_{k,i}^* := l$ to add cuts to the MP (cf., Section 5.3). Otherwise, if we reach state $l = |\omega_k|$ with $\bar{\zeta}(k,l) \ge 0$, a feasible tour has been found and we set $l_{k,i}^* := 0$. Furthermore, we save the solution to omit solving the subproblem again, if ω_k is part of a future solution to [MP].

The proposed procedure of determining the optimal deadheading routes for a given ω_k has at most $|\omega_k|$ states. In each state, the runtime for determining the optimal detour is asymptotically bounded by $O(s)$ (cf., Section 3.4), resulting in an asymptotic runtime of $O(|\omega_k|\cdot s)$ to solve the subproblem for a single schedule ω_k. This has to be repeated for all vehicles $k = 1,\dots,m$ and all $\omega_k \in \Omega_k$. The number of vehicles m cannot reasonably be greater than the number of trips n. Furthermore, $\sum_{k\in\{1,\dots,m\}} |\omega_k| \le 3\cdot n$ and $|\Omega_k| \le \frac{d^2-d}{2}$ holds. Hence, the feasibility status, detours, and charging breaks for a given master solution can be determined in at most $O(n\cdot s\cdot d^2)$ time.

The idea of the above procedure can also be applied in a reverse manner. It is sufficient for the vehicle to return to the final depot with exactly zero SOC. Starting from the final depot, i.e. state $l = |\omega_k|$, we can derive the minimum sufficient SOC at a previous state $l-1$ from the minimum sufficient SOC at state l. If we reach a state l where the minimum sufficient SOC exceeds $\hat{c}$, the assignment is infeasible, we save $l_{k,i}^{*,\text{rev}} = l$, and terminate the procedure. Otherwise, if we reach state $l = 0$, we set $l_{k,i}^{*,\text{rev}} := |\omega_k| + 1$. Clearly, concerning the feasibility of an assignment, the original and the reversed procedure yield the same result. However, $l_{k,i}^{*,\text{rev}}$ of the reverse procedure can be used for additional cuts if an assignment is infeasible. Hence, we only apply the reverse procedure, whenever the original procedure encounters an infeasible assignment.

5.3. Cuts

The subproblem may be infeasible in two distinctive ways, either due to the capacity limit of one or multiple depots being violated (i.e., $Z > 0$) or due to the working schedule $\hat{\omega}_k$ being impossible to execute feasibly for one or multiple vehicles $k = 1, \dots, m$ (i.e., $\min_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\} > 0$, $\exists k = 1, \dots, m$).

In the first case, we add the cut

$$1 \le \sum_{j\in J} \sum_{\substack{j'\in J:\\ \bar{x}_{j,j'}=1}} \left(1 - x_{j,j'}\right)$$

to the model [MP], to which we refer as *depot feasibility* cut. The cut ensures that at least one working schedule $\hat{\omega}_k$ for one vehicle $k \in \{1, \dots, m\}$ changes, making the current solution infeasible for model [MP] and thereby progressing the search.

In the second case, when there exists no feasible schedule ω_k for some vehicle k, we add so-called *schedule feasibility* cuts to model [MP] to exclude this infeasible solution and solutions sharing the same infeasible subset of consecutively executed trips from the search space. Formally, feasibility cuts are added for all $k \in \{1, \dots, m \mid \min_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\} > 0\}$. Since $\max_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\}$ is the first trip that could not be feasibly served anymore for any choice of initial depot, at least one of the first $\max_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\}$ trips must change to attain feasibility. We ensure this by adding the cut

$$1 \le \sum_{l=1}^{\max_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\}-1} \left(1 - x_{\hat{\omega}_k(l),\hat{\omega}_k(l+1)}\right)$$

to the constraint set of model [MP]. Likewise, if it is not possible to execute the trips $\max_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^{*,\text{ rev}}_{i,k}\right\}$ to $|\hat{\omega}_k|$ feasibly, we add the cut

$$1 \le \sum_{l=\min_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^{*,\text{ rev}}_{i,k}\right\}-1}^{|\hat{\omega}_k|-1} \left(1 - x_{\hat{\omega}_k(l),\hat{\omega}_k(l+1)}\right).$$

If $\min_{i\in\{1,\dots,|\Omega_k|\}}\left\{l^*_{i,k}\right\} = 0$, $\forall k = 1, \dots, m$, and $Z = 0$ the solution is feasible for the original problem and is stored as the currently best known solution. We note that the black-box solver passes every encountered integer solution that is feasible for model [MP] to be evaluated by solving the emerging subproblems. That means that even if the solution is feasible for the original problem, it is not necessarily optimal. Hence, a feasible solution presents merely an upper bound on [MP] and the original problem. Therefore, to expedite solving the master problem optimally, we add the *optimality* cut

$$\sum_{j\in J} x_{0,j} \le m - 1 \tag{34}$$

to [MP]. As soon as the solution space of [MP] becomes empty, the last found solution that is feasible to the original problem is guaranteed to also be optimal and the procedure terminates.

6. Computational study

In this section, we test the computational performance of our proposed branch-and-check procedure and compare it to a default solver, namely CPLEX, solving the proposed [EVSP-MD-FS] IP model. To do so, we generate randomized instances of various sizes and based on in-plant logistics. We describe the instance generation in the following section.

Furthermore, we derive some managerial insights into the influence of the battery capacity and charging mode on the fleet size. Finally, we investigate the influence of the number of warehouses and depots in the production facility.

6.1. Benchmark instances and computational environment

6.1.1. Generating instances for the EVSP-MD-FS based on an in-plant milk-run logistics setting

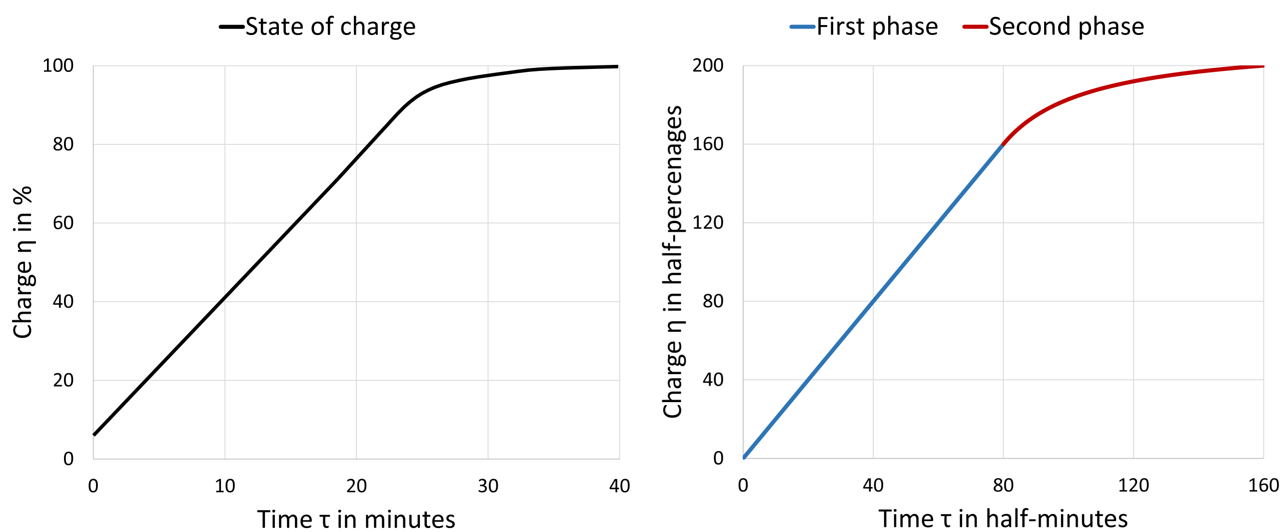
We model the instances for EVSP-MD-FS based on an in-plant milk-run logistics setting in accordance with the one regarded by Emde et al. (2018). As shown in Fig. 1a, we assume a facility where the production area and the warehouses are separated. This need not always be the case, but is a reasonable assumption for a lot of practical cases. We implement this assumption by placing the warehouses at random locations within a 1000 m × 1000 m area that represents the facility, where the production area is located at coordinates (1000 m, 500 m), i.e., centrally at the right border of the facility. All warehouses are assumed to have an internal size of 200 m × 200 m and the production area is set to 400 m × 400 m.

We further assume that the tow train depots are located within the warehouses and that all depots, and exclusively depots, can be used as charging stations. As described in Section 1, a trip starts in a warehouse, where the tow train is loaded with the materials required at the production stations. The tow train then sets off to the production area and visits a set of predetermined assembly cells to deliver the required goods. Once the final assembly cell has been supplied, the tow train is ready to process the next trip or return to a depot. This is modeled by placing the depots and the trips' start points at random locations within the warehouses, where each warehouse can host at most one depot. The trips' end points are set to random locations within the production area.

The proposed branch-and-check approach can handle real-valued parameters. However, the [EVSP-MD-FS] model requires all parameters to be integer to yield an exact result, which is why we round parameters to integer values in the following. Considering time parameters, we use a precision of half-minutes. I.e., an integer value of 1 represents half a minute. The trips' start times are determined according to Emde et al. (2018). The first trip is set so start at $s_1 = 10$ half-minutes, such that the trip's start can be reached by a tow train departing from a depot no sooner than time 0. The starting times of all consecutive trips are set iteratively according to $s_j = s_{j-1} + rnd([0, 10])$ half-minutes, where $rnd([0, 10])$ is a random integer from the interval [0, 10], which results in close to real-world timetables (Emde et al., 2018). Hence, on average, there are 24 tips per hour, which – for comparison – falls in the upper quarter of trips per hour reported for tow train systems in the German industry (cf., Lieb, Klenk, Galka, & Keuntje, 2017). The trips' execution times are set to random integers drawn from the interval [20, 50] half-minutes, since those are typical trip durations in practice. A trip's end time is its start time plus its execution time.

To determine realistic time requirements for the detours, we first calculate respective distances applying the rectilinear metric. The total distance between two locations (e.g., the end point of a trip and the location of a depot) is set to be the sum of the internal distance (i.e., the distance within the warehouse and production area) and the external distance (i.e., the distance between warehouses and the production area), which is in line with our assumed facility layout. The travel times for all detours are then calculated by dividing the respective distances by an assumed travel speed of 10 kph and rounding to the nearest half-minute value, where 10 kph is a representative speed for tow trains.

We state all battery-related values as percentages of the maximum charge and use a precision of half-percentages. I.e., an integer value of 1 represents half a percentage of the maximum SOC. The trips' battery requirements may not be strictly proportional to their distance or execution time, since each trip requires a different number of stops at assembly stations and the tow train may need to carry different loads on every trip. To account for this fact, we determine the trips' battery requirements in the following way. First, we associate the minimum trip duration of 20 half-minutes



(a) Exemplary lithium-ion battery charging function according to Kisacikoglu et al. (2011).

(b) Tow train battery charging function.

Fig. 3. Non-linear battery charging functions.

with a battery requirement of 12.5% and the maximum trip duration of 50 half-minutes with 30%, which is based on the technical data of a representative tow train (cf., Still, 2019). Second, we interpolate each trip's preliminary battery requirement based on its duration. Third, we randomize the battery requirement by multiplying the preliminary battery requirement with a random factor from the interval $[0.75, 1.25]$ and round it to the nearest half-percentage value. While processing detours, the tow train is mostly empty and does not need to stop frequently. Hence, here, distance, execution time and battery requirement are proportional in good approximation. Therefore, we calculated the detours' battery requirements by multiplying their respective distances with the factor $5\frac{\%}{\text{km}}$, which, again, is derived from a representative tow train's technical data (cf., Still, 2019).

Using the proposed scheme, we generate ten instances of each size $n \in \{50, 100, 200\}$ with two, three, and five warehouses, respectively, where every warehouse also hosts a depot. Note that with $n = 200$, an instance covers roughly an eight-hour workday. Moreover, note that at this point, we do not make any assumptions regarding the way in which the batteries are charged. This is discussed in Section 6.1.2. All instances are named according to the scheme "I-*s*-*d*-*n*-##", where "##" is a continuous counting number.

6.1.2. Modeling (piecewise) non-linear and constant-time battery charging

Both the [EVSP-MD-FS] model and the proposed branch-and-check approach are able to handle various battery charging functions. In our computational study, we consider two ways of realistic battery charging, namely constant-time battery charging, e.g., battery swapping, and (piecewise) concave non-linear charging as common for lithium-ion batteries, which are frequently used for powering electric vehicles in intra-logistic applications.

As stated in Section 3.1, battery swapping (or other means of constant-time battery charging) can be modeled by setting r, the time between the arrival of the vehicle at the charging station and the start of the actual charging, to the time it takes to swap the battery. The battery charging function is $\eta(\tau) = \hat{c}$ in this case. Based on our observations, the time it takes to swap a tow train's battery can vary depending on the skills of the worker, the setup of the charging station and, most importantly, the exact model of the tow train. Nevertheless, we found three minutes to be a representative value, for a broad variety of cases. Hence, we set $r = 6$ half-minutes.

Lithium-ion batteries are a common type of batteries frequently used to power electric vehicles (Kisacikoglu, Ozpineci, & Tolbert, 2011; Pelletier et al., 2017). Their charging function is (piecewise) non-linear, concave, and usually consists of two phases, as exemplarily depicted in Fig. 3a and explained in Section 1. To briefly recap, first, between 0% of total charge up to about 75% to 90%, depending on the exact battery type, charging is linear with time τ. In the second phase, the charging rate continuously depreciates, which results in a charging function that gradually approaches 100% over time. The transition between both phases is continuous. Depending on the exact type of battery, the first phase takes about 25% to 75% of the total charging time and the second phase takes between 75% to 25% of the time (Kisacikoglu et al., 2011; Pelletier et al., 2017). In the following, we assume that both phases take an equal duration and linear charging ends at 80% of total charge. Furthermore, we approximate the charging function in the second phase with a function of the general shape of $\eta^{\text{second phase}}(\tau) = k_1 + \frac{k_2}{k_3+x}$, where k_1, k_2 and k_3 are adjustable factors. Given these assumption and the battery's charging rate in the first phase, we can determine the factors k_1, k_2 and k_3 by fitting $\eta^{\text{second phase}}(\tau)$ to the end of the first phase with a continuous transition from the linear charging rate. We assume a battery charging rate in the linear phase of $2\frac{\%}{\text{min}}$, which is about an average value for most tow-trains (cf. Emde et al., 2018). This results in the battery charging function

$$\eta(\tau) = \begin{cases} 2 \cdot \tau & \text{if } \tau \leq 80 \\ \frac{640}{3} - \frac{\frac{12800}{9}}{\tau - \frac{160}{3}} & \text{if } 80 < \tau \leq 160, \\ 200 & \text{if } 160 < \tau \end{cases}$$

where τ is in the unit of half-minutes and $\eta(\tau)$ is in the unit of half-percentages. The charging function is depicted in Fig. 3b. Furthermore, we assume the delay between the arrival at the station and the beginning of the charging procedure to be $r = 1$ half-minute.

6.2. Computational results

In this section, we present our computational results. All testing was performed on an Intel Core i7-6700 CPU @ 3.40 Gigahertz and with 16 Gigabyte of RAM. All algorithms were implemented in C# and CPLEX (version 12.10) was used as default solver (for the [EVSP-MD-FS] IP as well as the [MP] master model and [Depot-Matching] subproblem model of the proposed branch-and-check procedure) at default settings. Furthermore, we limited the runtime to solve a single instance to 3600 seconds (i.e., 1 h). The results of the tests are summarized in the following. Detailed reports on the results as well as the original instances are provided as online supplementary material at https://doi.org/10.5281/zenodo.6406201.

6.2.1. Performance evaluation

We solved all instances with CPLEX and with the proposed BCH approach. The results are summarized in Table 4. Generally, CPLEX performed better for constant-time charging than for non-linear charging. It was able to solve all instances with $n \leq 100$ to optimality for both types of battery charging within a runtime of 3600 s. For instances with $n = 200$, CPLEX was able to find solutions, which were not always optimal, however. Especially for non-linear battery charging, the respective optimality gaps were rather large with 67% on average.

The proposed BCH procedure clearly outperformed CPLEX. For constant-time charging, the BCH procedure solved all instances in at most 2.6 s. For non-linear charging, the BCH procedure solved all instances with $n = 50$ in below 0.1 s and every instance with $n = 100$ in no more than 5.1 s. While the BCH procedure could solve all T-3-3-200-## instances in 364.8 s on average, one instance required 3418.4 s to be solved while the other instances were solved much faster.

Our results clearly indicate that instances were easier to solve if constant-time charging was assumed, especially if instances were large. We attribute this to constant-time charging being generally quicker than non-linear charging, which makes it easier to find feasible solutions for the former. This becomes evident when the “s. f. cut” column in Tables 4 is considered. It indicates an instance's hardness by stating the number of times the MP's solution was infeasible (due to violated battery constraints) such that one or multiple schedule feasibility cuts had to be added to the MP. The observation is also supported by the results in Section 6.2.4, where instances with less battery capacity were harder to solve.

6.2.2. Performance on instances with additional constraints

As described in Section 3, we also consider optional constrains for EVSP-MD-FS, namely that vehicles must return to their depot of initial departure and that the depots' capacities are limited. While these constraints are not currently relevant in the production plants we visited, they are often discussed in the VSP literature and may also play a role in intra-logistics, especially in such cases where the depots are small and vehicles are fixedly assigned to them. To evaluate the effects of these constraints, we generated constrained instances from regular instances as follows. From each regular instance in Table 4, we constructed a constrained instance with the same time and battery parameters. To obtain reasonable capacity limits for the depots, we took the regular instance's optimal objective value, multiplied it by a factor of 1.25, rounded the result up, and split it into integer values that we distributed randomly between the depots, where we made sure that each depot has at least a capacity of 1. Moreover, we made the instance require that vehicles must return to their depot of initial departure. Each constrained instance is labeled according to the regular instance it is based on except for an initial “C” instead of “I”.

We solved all constrained instances with CPLEX and with the proposed BCH approach. The results are summarized in Table 5.

The results indicate that the constrained instances were generally harder to solve for both CPLEX and – to a lesser degree – the proposed BCH approach. Especially for the former, required runtimes increase significantly such that for $n = 200$ and non-linear charging, CPLEX did not find any solution at all within the runtime limit. On the other hand, runtimes only increased slightly for the BCH approach, such that all instances except for instances C-5-5-200-02 at non-linear charging were solved optimally within the runtime limit. This is in line with the rather low number of applied depot feasibility cuts (cf., the “d. f. cut” columns in Table 5), which never exceeded two.

6.2.3. Performance on large instances

In Section 6.2.1, we considered instances with up to $n = 200$ trips, which cover roughly an eight-hour shift. To test the performance of our solution method on even larger instances and see how it scales from a theoretical perspective, we apply the scheme described in Section 6.1.1 to randomly generate instances with $s = d = 5$ as well as 1000, 2000, 3000, and 4000 trips, which is well beyond the number of trips we expect in practical intra-logistic applications.

We tried to solve the respective instances by applying CPLEX to the IP model. However, during initialization of the model, CPLEX always reported an out of memory error. For the BCH approach, we set a runtime limit of 24 h (86400 s). The results are given in Table 6.

For non-linear charging, only exceptionally bad solutions were found within the runtime limit. We explain this as follows. At the beginning of the procedure, CPLEX, which is used to solve the MP in the BCH approach, forwards rather bad solutions to the subproblem, which (unsurprisingly) are validated to be feasible. Afterwards, CPLEX repeatedly forwards solutions with objective values equaling the lower bound to the subproblem, which are found to be infeasible, however. As a consequence, no close-to-optimal solutions are explored within the runtime and the best found solutions remain at a bad objective value.

On the other hand, for constant-time charging, the BCH approach was able to find optimal solutions for all instances except instance I-5-5-4000-2, where the optimality gap is only 1. This is quite a remarkable result for an exact solution approach and indicates that our approach may be quite suitable in practice even for very large instances if battery-swapping technology is used.

Note that, as mentioned in Section 6.2, all computational tests were performed on a desktop PC with 16 GB of RAM available. However, for the larger instances, the BCH procedure required far greater memory, which sometimes even exceeded 50 GB during processing. Hence, hard drive memory was used, which slowed down processing significantly. For example, for the I-5-5-4000-## instances, reading out an integer candidate solution $\bar{\boldsymbol{x}}$ from the MP required more than half an hour, even though the number of variables is bounded by $O(n^2)$. Since we attribute this time to the subproblem's runtime, it also explains the large proportion of runtime spent solving the subproblem in Table 6. Hence, if the BCH would be executed on a machine with sufficient RAM, we expect further performance increases.

6.2.4. Influence of the battery capacity and charging mode

This section investigates the influence of the battery capacity and the charging mode on the number of required vehicles. We examined the effects of increasing and decreasing the battery capacity by 10%, 20% and 30% and compared it to the status quo. Changing the battery capacity requires adjusting the battery charging function $\eta(\tau)$, which we did by simply multiplying its output by the same factor. We performed the experiment for all instances with $n = 100$. The results of the experiment are summarized in Table 7 and Fig. 4.

The battery charging mode did not influence the number of required vehicles for the regular or an increased battery capacity.

Table 4
Computational test on the I-$|S|-|D|-n$-## instances.

instance	non-linear battery recharging											constant-time battery recharging										
	CPLEX			branch-and-check								CPLEX			branch-and-check							
lable	value	LB	time (in s)	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts	value	LB	time (in s)	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts
I-2-2-50-01	13	13	9.5	13	13	0.1	89.1	10.9	1	0	1	13	13	7.3	13	13	0.0	100.0	0.0	0	0	1
I-2-2-50-02	12	12	11.4	12	12	0.1	86.3	13.7	11	0	1	12	12	6.9	12	12	0.0	100.0	0.0	0	0	1
I-2-2-50-03	12	12	9.4	12	12	0.0	97.6	2.4	4	0	1	12	12	8.2	12	12	0.0	100.0	0.0	0	0	1
I-2-2-50-04	17	17	8.6	17	17	0.0	93.8	6.3	0	0	1	17	17	6.1	17	17	0.0	93.8	6.3	0	0	1
I-2-2-50-05	15	15	8.2	15	15	0.0	97.1	2.9	2	0	1	15	15	6.4	15	15	0.0	94.7	5.3	1	0	1
I-2-2-50-06	13	13	9.2	13	13	0.0	94.6	5.4	4	0	1	13	13	6.2	13	13	0.0	100.0	0.0	0	0	1
I-2-2-50-07	16	16	8.8	16	16	0.0	100.0	0.0	0	0	1	16	16	6.1	16	16	0.0	100.0	0.0	0	0	1
I-2-2-50-08	12	12	9.8	12	12	0.0	95.1	4.9	8	0	1	12	12	6.3	12	12	0.0	100.0	0.0	0	0	1
I-2-2-50-09	16	16	8.3	16	16	0.0	95.0	5.0	1	0	1	16	16	6.6	16	16	0.0	93.8	6.3	0	0	1
I-2-2-50-10	13	13	11.5	13	13	0.0	95.0	5.0	0	0	1	13	13	6.2	13	13	0.0	100.0	0.0	0	0	1
mean	**13.9**	**13.9**	**9.5**	**13.9**	**13.9**	**0.0**	**94.4**	**5.6**	**3.1**	**0.0**	**1.0**	**13.9**	**13.9**	**6.6**	**13.9**	**13.9**	**0.0**	**98.2**	**1.8**	**0.1**	**0.0**	**1.0**
I-3-3-100-01	15	15	342.3	15	15	3.1	68.2	31.8	538	0	4	15	15	66.7	15	15	0.1	94.3	5.8	0	0	1
I-3-3-100-02	17	17	76.8	17	17	0.2	75.3	24.7	13	0	1	17	17	85.7	17	17	0.1	89.0	11.0	0	0	1
I-3-3-100-03	16	16	200.7	16	16	0.3	75.1	24.9	13	0	1	16	16	75.9	16	16	0.2	91.1	8.9	1	0	1
I-3-3-100-04	16	16	265.7	16	16	0.3	75.4	24.6	17	0	1	16	16	54.4	16	16	0.1	85.1	14.9	0	0	2
I-3-3-100-05	18	18	450.7	18	18	5.1	64.5	35.5	824	0	3	18	18	39.8	18	18	0.1	90.9	9.1	1	0	1
I-3-3-100-06	15	15	1464.3	15	15	4.1	68.6	31.4	824	0	4	15	15	138.8	15	15	0.1	93.1	6.9	0	0	1
I-3-3-100-07	14	14	3597.4	14	14	0.8	68.4	31.6	77	0	3	14	14	99.0	14	14	0.1	94.4	5.6	0	0	1
I-3-3-100-08	15	15	478.3	15	15	1.8	59.3	40.8	204	0	4	15	15	146.7	15	15	0.1	91.0	9.0	2	0	1
I-3-3-100-09	19	19	117.4	19	19	0.2	65.4	34.6	8	0	1	19	19	26.2	19	19	0.1	87.7	12.4	0	0	1
I-3-3-100-10	19	19	267.7	19	19	0.4	63.1	36.9	21	0	4	19	19	42.7	19	19	0.1	89.4	10.6	1	0	1
mean	**16.4**	**16.4**	**726.1**	**16.4**	**16.4**	**1.6**	**68.3**	**31.7**	**253.9**	**0.0**	**2.6**	**16.4**	**16.4**	**77.6**	**16.4**	**16.4**	**0.1**	**90.6**	**9.4**	**0.5**	**0.0**	**1.1**
I-5-5-200-01	20	18	3600.0	18	18	7.2	53.7	46.4	96	0	3	19	18	3600.0	18	18	0.4	84.1	15.9	0	0	1
I-5-5-200-02	36	17	3600.0	17	17	3418.4	86.4	13.6	38,018	0	4	17	17	1039.2	17	17	0.8	87.4	12.6	1	0	1
I-5-5-200-03	37	15	3600.0	15	15	1.7	58.0	42.0	34	0	1	15	15	2913.9	15	15	0.4	87.7	12.3	0	0	1
I-5-5-200-04	17	16	3600.0	16	16	1.3	70.2	29.8	10	0	1	16	16	1750.3	16	16	2.6	42.4	57.6	10	0	3
I-5-5-200-05	18	18	1439.9	18	18	8.8	47.1	52.9	76	0	3	18	18	1750.9	18	18	0.9	82.5	17.5	1	0	1
I-5-5-200-06	16	16	1904.1	16	16	1.1	82.6	17.4	5	0	1	16	16	1037.1	16	16	0.4	84.9	15.1	0	0	1
I-5-5-200-07	31	16	3600.0	16	16	5.8	63.5	36.5	54	0	3	16	16	344.5	16	16	0.4	85.3	14.7	0	0	1
I-5-5-200-08	47	16	3600.0	16	16	2.7	56.2	43.8	34	0	2	16	16	2121.4	16	16	0.4	85.7	14.4	0	0	1
I-5-5-200-09	18	18	1365.1	18	18	191.7	45.7	54.3	4331	0	4	18	18	2250.0	18	18	0.8	91.5	8.5	0	0	1
I-5-5-200-10	41	18	3600.0	18	18	9.7	52.2	47.8	156	0	3	18	18	1486.8	18	18	0.4	85.0	15.0	0	0	1
mean	**28.1**	**16.8**	**2990.9**	**16.8**	**16.8**	**364.8**	**61.6**	**38.4**	**4281.4**	**0.0**	**2.5**	**16.9**	**16.8**	**1829.4**	**16.8**	**16.8**	**0.8**	**81.6**	**18.4**	**1.2**	**0.0**	**1.2**

LB = lower bound at the point of termination; s./ d. inf. cuts = number of added schedule/ depot infeasibility cuts; opt. cuts = number of added optimality cuts; time MP/ SP = relative runtime of the BCH master problem/ of the BCH subproblem including cut generation

Table 5
Computational test on the C-$|S|-|D|-n$-## instances.

instance	non-linear battery recharging											constant-time battery recharging										
	CPLEX			branch-and-check								CPLEX			branch-and-check							
lable	value	LB	time (in s)	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts	value	LB	time (in s)	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts
C-2-2-50-01	13	13	10.2	13	13	0.0	81.6	18.4	1	0	1	13	13	7.1	13	13	0.0	82.6	17.4	0	0	1
C-2-2-50-02	12	12	13.1	12	12	0.0	83.3	16.7	11	0	1	12	12	9.0	12	12	0.0	72.0	28.0	0	0	1
C-2-2-50-03	12	12	11.0	12	12	0.0	78.4	21.6	4	0	1	12	12	7.7	12	12	0.0	73.7	26.3	0	0	1
C-2-2-50-04	17	17	8.7	17	17	0.0	79.0	21.1	0	0	1	17	17	6.3	17	17	0.0	79.0	21.1	0	0	1
C-2-2-50-05	15	15	8.8	15	15	0.0	85.7	14.3	2	0	1	15	15	6.2	15	15	0.0	77.3	22.7	1	0	1
C-2-2-50-06	13	13	9.3	13	13	0.0	83.3	16.7	4	0	1	13	13	6.6	13	13	0.0	76.2	23.8	0	0	1
C-2-2-50-07	16	16	8.7	16	16	0.0	79.0	21.1	0	0	1	16	16	6.3	16	16	0.0	76.2	23.8	0	0	1
C-2-2-50-08	12	12	10.4	12	12	0.0	84.2	15.8	8	0	1	12	12	8.4	12	12	0.0	83.3	16.7	0	0	1
C-2-2-50-09	16	16	9.4	16	16	0.0	71.4	28.6	1	0	1	16	16	6.0	16	16	0.0	75.0	25.0	0	0	1
C-2-2-50-10	13	13	12.4	13	13	0.0	77.3	22.7	0	0	1	13	13	6.5	13	13	0.0	77.3	22.7	0	0	1
mean	**13.9**	**13.9**	**10.2**	**13.9**	**13.9**	**0.0**	**80.3**	**19.7**	**3.1**	**0.0**	**1.0**	**13.9**	**13.9**	**7.0**	**13.9**	**13.9**	**0.0**	**77.2**	**22.8**	**0.1**	**0.0**	**1.0**
C-3-3-100-01	15	15	1450.1	15	15	4.8	65.1	34.9	987	2	3	15	15	48.2	15	15	0.1	88.0	12.1	0	0	1
C-3-3-100-02	17	17	818.6	17	17	0.2	68.2	31.8	13	0	1	17	17	54.4	17	17	0.1	86.6	13.4	0	0	1
C-3-3-100-03	16	16	272.6	16	16	0.3	73.4	26.6	13	0	1	16	16	48.3	16	16	0.1	86.6	13.4	1	0	1
C-3-3-100-04	16	16	1620.2	16	16	0.2	70.3	29.7	17	0	1	16	16	87.9	16	16	0.1	75.0	25.0	0	0	2
C-3-3-100-05	18	18	1008.9	18	18	4.7	63.8	36.2	824	0	3	18	18	98.5	18	18	0.1	87.5	12.5	1	0	1
C-3-3-100-06	15	15	988.0	15	15	1.9	62.2	37.8	374	2	1	15	15	65.1	15	15	0.1	88.1	11.9	0	0	1
C-3-3-100-07	14	14	1766.6	14	14	0.9	57.1	42.9	98	2	1	14	14	52.3	14	14	0.1	85.7	14.3	0	0	1
C-3-3-100-08	15	15	1738.5	15	15	8.0	61.0	39.0	1140	2	3	15	15	114.5	15	15	0.1	87.3	12.7	2	0	1
C-3-3-100-09	19	19	945.1	19	19	0.2	66.7	33.3	8	0	1	19	19	44.1	19	19	0.1	84.6	15.4	0	0	1
C-3-3-100-10	19	19	272.5	19	19	0.4	60.6	39.4	18	2	1	19	19	28.5	19	19	0.1	86.1	13.9	1	0	1
mean	**16.4**	**16.4**	**1088.1**	**16.4**	**16.4**	**2.2**	**64.8**	**35.2**	**349.2**	**1.0**	**1.6**	**16.4**	**16.4**	**64.2**	**16.4**	**16.4**	**0.1**	**85.6**	**14.4**	**0.5**	**0.0**	**1.1**
C-5-5-200-01	-	18	3600.0	18	18	13.2	49.1	50.9	163	2	1	23	18	3600.0	18	18	0.4	83.5	16.5	0	0	1
C-5-5-200-02	-	17	3600.0	18	17	3600.0	90.4	9.6	26,417	2	2	23	17	3600.0	17	17	0.8	85.8	14.2	1	0	1
C-5-5-200-03	-	15	3600.0	15	15	1.8	59.7	40.4	34	0	1	-	15	3600.0	15	15	0.4	85.4	14.7	0	0	1
C-5-5-200-04	-	16	3600.0	16	16	1.3	70.5	29.5	10	0	1	-	16	3600.0	16	16	2.6	40.8	59.2	10	2	1
C-5-5-200-05	-	18	3600.0	18	18	36.3	38.1	61.9	509	2	2	22	18	3600.0	18	18	0.9	83.1	16.9	1	0	1
C-5-5-200-06	-	16	3600.0	16	16	1.1	81.9	18.1	5	0	1	-	16	3600.0	16	16	0.4	84.3	15.7	0	0	1
C-5-5-200-07	-	16	3600.0	16	16	11.2	58.5	41.5	139	2	1	21	16	3600.0	16	16	0.4	83.5	16.5	0	0	1
C-5-5-200-08	-	16	3600.0	16	16	2.8	56.5	43.5	34	0	2	19	16	3600.0	16	16	0.4	85.1	14.9	0	0	1
C-5-5-200-09	-	18	3600.0	18	18	17.5	41.0	59.0	273	2	2	23	18	3600.0	18	18	0.8	89.9	10.1	0	0	1
C-5-5-200-10	-	18	3600.0	18	18	15.6	54.3	45.7	285	2	1	24	18	3600.0	18	18	0.4	83.5	16.5	0	0	1
mean	**-**	**16.8**	**3600.0**	**16.9**	**16.8**	**370.1**	**60.0**	**40.0**	**2786.9**	**1.2**	**1.4**	**22.1**	**16.8**	**3600.0**	**16.8**	**16.8**	**0.8**	**80.5**	**19.5**	**1.2**	**0.2**	**1.0**

LB = lower bound at the point of termination; s./ d. inf. cuts = number of added schedule/ depot infeasibility cuts; opt. cuts = number of added optimality cuts; time MP/ SP = relative runtime of the BCH master problem/ of the BCH subproblem including cut generation; - = no feasible solution was found within the runtime

Table 6
Computational test on large instances.

instance	non-linear charging								constant-time charging							
lable	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts	value	LB	time (in s)	time MP (in %)	time SP (in %)	s. inf. cuts	d. inf. cuts	opt. cuts
I-5-5-1000-01	291	18	86400.0	47.8	52.2	61,044	0	2	18	18	46.6	78.1	21.9	0	0	1
I-5-5-1000-02	291	17	86400.0	62.5	37.5	51,369	0	2	17	17	86.0	61.3	38.7	5	0	1
I-5-5-1000-03	298	19	86400.0	39.3	60.7	57,756	0	2	19	19	46.0	77.5	22.5	0	0	1
I-5-5-1000-04	295	18	86400.0	45.6	54.5	69,618	0	2	18	18	65.2	58.2	41.8	2	0	1
I-5-5-1000-05	284	18	86400.0	40.3	59.7	69,693	0	2	18	18	37.2	68.1	31.9	0	0	1
mean	**291.8**	**18.0**	**86400.0**	**47.1**	**52.9**	**61896.0**	**0.0**	**2.0**	**18.0**	**18.0**	**56.2**	**68.6**	**31.4**	**1.4**	**0.0**	**1.0**
I-5-5-2000-01	596	16	86400.0	54.9	45.1	2206	0	2	16	16	1043.0	59.1	41.0	3	0	1
I-5-5-2000-02	583	19	86400.0	45.6	54.4	4316	0	2	19	19	2777.2	33.4	66.7	10	0	2
I-5-5-2000-03	577	18	86400.0	60.6	39.5	3158	0	2	18	18	4766.5	32.3	67.7	23	0	2
I-5-5-2000-04	576	21	86400.0	28.7	71.3	4860	0	2	21	21	263.3	49.4	50.6	0	0	1
I-5-5-2000-05	578	18	86400.0	59.9	40.1	3616	0	2	18	18	665.2	84.6	15.4	0	0	1
mean	**582.0**	**18.4**	**86400.0**	**49.9**	**50.1**	**3631.2**	**0.0**	**2.0**	**18.4**	**18.4**	**1903.1**	**51.8**	**48.2**	**7.2**	**0.0**	**1.4**
I-5-5-3000-01	877	21	86400.0	9.8	90.3	46	0	2	21	21	2174.5	77.3	22.7	0	0	1
I-5-5-3000-02	872	18	86400.0	13.6	86.4	47	0	2	18	18	1714.9	77.0	23.1	0	0	1
I-5-5-3000-03	876	20	86400.0	16.9	83.1	60	0	2	20	20	2943.9	84.9	15.1	0	0	1
I-5-5-3000-04	883	19	86400.0	12.1	87.9	86	0	2	19	19	2239.4	64.9	35.1	2	0	1
I-5-5-3000-05	879	22	86400.0	11.4	88.6	54	0	2	22	22	1132.8	50.0	50.0	0	0	1
mean	**877.4**	**20.0**	**86400.0**	**12.8**	**87.2**	**58.6**	**0.0**	**2.0**	**20.0**	**20.0**	**2041.1**	**70.8**	**29.2**	**0.4**	**0**	**1**
I-5-5-4000-01	4000	18	86400.0	7.1	92.9	19	0	1	18	18	4211.9	78.2	21.8	0	0	1
I-5-5-4000-02	4000	19	86400.0	10.5	89.5	19	0	1	20	19	86400.0	54.7	45.3	348	0	2
I-5-5-4000-03	4000	17	86400.0	9.7	90.3	36	0	1	17	17	7106.2	89.1	10.9	0	0	1
I-5-5-4000-04	4000	20	86400.0	5.2	94.8	6	0	1	20	20	4857.9	53.9	46.1	1	0	1
I-5-5-4000-05	4000	19	86400.0	10.0	90.0	6	0	1	19	19	12185.7	67.0	33.0	3	0	1
mean	**4000.0**	**18.6**	**86400.0**	**8.5**	**91.5**	**17.2**	**0.0**	**1.0**	**18.8**	**18.6**	**22952.3**	**68.6**	**31.4**	**70.4**	**0.0**	**1.2**

LB = lower bound at the point of termination; s./ d. inf. cuts = number of added schedule/ depot infeasibility cuts; opt. cuts = number of added optimality cuts; time MP/ SP = relative runtime of the BCH master problem/ of the BCH subproblem including cut generation.

Table 7
Computational study on the influence of the battery capacity and charging.

instance	non-linear with a capacity of							constant-time with a capacity of						
label	70%	80%	90%	100%	110%	120%	130%	70%	80%	90%	100%	110%	120%	130%
I-3-3-100-01	^{5}20	^{3}18	^{1}16	15	15	15	15	15	15	15	15	15	15	15
I-3-3-100-02	^{3}20	17	17	17	17	17	17	17	17	17	17	17	17	17
I-3-3-100-03	^{4}20	^{1}17	16	16	16	16	16	16	16	16	16	16	16	16
I-3-3-100-04	^{3}19	16	16	16	16	16	16	16	16	16	16	16	16	16
I-3-3-100-05	^{4}23	^{3}21	^{1}19	18	18	18	18	19	18	18	18	18	18	18
I-3-3-100-06	^{6}21	^{3}18	^{1}16	15	15	15	15	^{1}16	15	15	15	15	15	15
I-3-3-100-07	^{4}18	^{2}16	^{1}15	14	14	14	14	14	14	14	14	14	14	14
I-3-3-100-08	^{2}18	^{2}17	15	15	15	15	15	16	15	15	15	15	15	15
I-3-3-100-09	^{1}20	19	19	19	19	19	19	19	19	19	19	19	19	19
I-3-3-100-10	^{3}22	19	19	19	19	19	19	19	19	19	19	19	19	19
mean	**20.1**	**17.8**	**16.8**	**16.4**	**16.4**	**16.4**	**16.4**	**16.7**	**16.4**	**16.4**	**16.4**	**16.4**	**16.4**	**16.4**

Preceding superscripts denote duality gaps >0 after 3600 s of runtime.

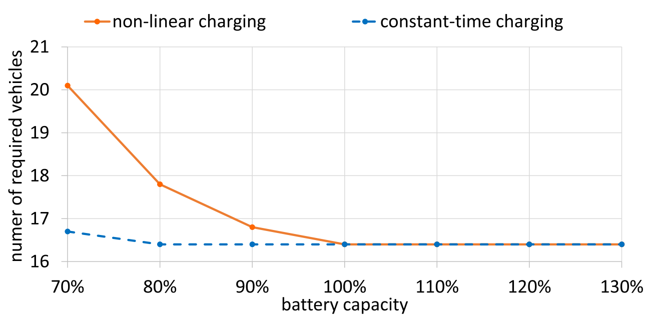


Fig. 4. Influence of the battery capacity and charging mode on the fleet size.

However, for below 100% of battery capacity, more vehicles were required for non-linear charging. For 70% of battery capacity, the required fleet size also slightly increased for constant-time charging. Furthermore, the lower the battery capacity, the harder it was for the BCH approach to find optimal solutions, especially for the case of non-linear charging.

Generally, non-linear battery charging required a larger fleet size than constant-time charging. The difference was only relevant for battery capacities below 100%, however. In conclusion, our experiments show that the battery charging mode and capacity did influence the required fleet size. Above a certain threshold, which was at about 90% to 100% battery capacity in our experiments, the effects became marginal, however.

Despite the generally low impact of the charging mode on the fleet size, it has an important influence. As the schedule infeasibility cuts in Tables 4 and 5 show, significantly more solutions needed to be evaluated for non-linear charging than for constant-time charging before the optimal solution was found. This strongly suggests that the solution space (i.e., the number of solutions with minimal fleet size) is much narrower for the former than for the latter. Consequently, if secondary objectives, additional constraints, or robustness aspects become relevant, constant-time charging offers greater flexibility.

As a final remark, we checked whether solutions found for either of the charging modes would be feasible for the other. While all solutions found for non-linear charging were also feasible for constant-time charging, the opposite was only true for nine solutions and only when the battery capacity was at 120% or above. This is in line with the findings of Olsen and Kliewer (2018) and Olsen and Kliewer (2020) inasmuch as it demonstrates the necessity to model plug-in charging as non-linear charging to avoid schedules that are infeasible in practice.

6.2.5. Influence of the number of depots

Our instances consider a facility with multiple warehouses (as depicted in Fig. 1a), where each warehouse has a separate tow train depot (which is also a charging station). In this section, we investigate how the number of warehouses and depots effects the fleet size.

We generated instances with $s \in \{1, 2, 3, 4\}$ from the regular I-5-5-200-## instances in the following way. Starting with a I-5-5-200-## instance with five warehouses, we merged the first and second warehouse into a single warehouse to attain a comparable instance with four warehouses. Likewise, to attain an instance with three warehouses, we merged warehouses one and two as well as three and four. To attain instances with two warehouses, we merged warehouses one, two, and five, as well as three and four. Finally, instances with a single warehouse were generated by merging all five warehouses. Whenever we merged two or more warehouses, we set the location of the resulting warehouse within the facility to the average location of all merged warehouses. We did the same for the location of the depot within the merged warehouse. Furthermore, we scaled up the size of the merged warehouse – and all trip start locations within –, such that the merged warehouse's area equals the sum of the separate warehouses' areas. All other parameters were left unchanged to get fair comparisons.

We solved all I-s-d-200-## instances for non-linear and constant-time battery charging with the BCH approach. The results are summarized in Table 8.

In general, our experiment only found a small influence in the number of warehouses and depots on the required fleet size. On average having a medium number of warehouses – three to four in our experiments – resulted in the smallest average fleet size. Having five decentralized or only one or two centralized warehouses resulted in marginally larger fleet sizes.

While, on average, detours increased for the case of fewer and therefore larger warehouses, they increased only by a comparatively small amount. On the other hand, since (to make the comparison fair) we assumed merged warehouses to be the average distance of the individual warehouses away from the production area, very long tours that require a large amount of charge were less crucial, which increased flexibility. According to our results, the trade-off was most favorable for a medium number of warehouses. Overall, the effect of the centralization was comparatively small, however, and we expect the warehouses' locations (as opposed to their number) to be a much greater factor in practice.

Table 8
Computational study on the influence of the number of depots/ warehouses.

instance	non-linear battery charging					constant-time battery charging				
label	$s=d=5$	$s=d=4$	$s=d=3$	$s=d=2$	$s=d=1$	$s=d=5$	$s=d=4$	$s=d=3$	$s=d=2$	$s=d=1$
I-s-d-200-01	18	18	18	18	18	18	18	18	18	18
I-s-d-200-02	17	*18	18	18	18	17	17	18	18	18
I-s-d-200-03	15	15	15	15	15	15	15	15	15	15
I-s-d-200-04	16	16	16	16	17	16	16	16	16	17
I-s-d-200-05	18	18	17	17	19	18	18	17	17	19
I-s-d-200-06	16	16	16	16	16	16	16	16	16	16
I-s-d-200-07	16	15	15	*16	16	16	15	15	15	16
I-s-d-200-08	16	16	17	17	17	16	16	17	17	17
I-s-d-200-09	18	18	18	18	*19	18	18	18	18	18
I-s-d-200-10	18	17	17	18	18	18	17	17	18	18
mean	**16.8**	**16.7**	**16.7**	**16.9**	**17.3**	**16.8**	**16.6**	**16.7**	**16.8**	**17.2**

* optimality was not proven within 3600 s

7. Conclusion

In this paper, we consider the multi-depot electric vehicle scheduling problem with the objective of minimizing the required fleet size, which is motivated by a novel application in in-plant logistics. We consider the problem for various battery charging functions, but primarily focus on realistic concave non-linear lithium-ion battery charging and constant-time charging (i.e., battery swapping) in the course of the paper. We briefly discuss the problem's computational complexity and show that finding an optimal solution is NP-hard.

For a discrete version of the problem, we formulate an IP model. For the real-valued (non-discrete) problem, an exact branch-and-check procedure is developed. The branch-and-check procedure decomposes the problem into two parts: first, a master problem with relaxed battery constraints that is concerned with assigning trips to vehicles and, second, a subproblem that schedules charging breaks and checks whether the master problem's solutions are feasible with regard to the battery constraints.

Computational tests show that solving the IP model with an off-the-shelf standard solver (i.e., CPLEX) results in a subpar performance. On the other hand, the branch-and-check procedure solves instances with up to $n=200$ trips in below 2.6 s and instances with up to $n=4000$ in a couple of hours on average, if constant-time charging is assumed. For non-linear charging, the performance is somewhat worse. Nevertheless, for instances with $n=200$ trips, the branch-and-check procedure is still able to find optimal solutions within a runtime of one hour. Based on our computational tests, we further derive the following take-home insights from a managerial viewpoint:

- Generally, constant-time battery charging requires a smaller fleet size than non-linear charging. This is especially relevant if the vehicles' maximum battery capacity is small. For larger battery capacities, the effect is marginal. Moreover, constant-time battery charging results in a larger solution space (i.e., more possible solutions with the minimal fleet size) and therefore generally more flexibility.
- Applying a model with constant-time charging to a situation with non-linear charging (i.e., plug-in charging) is not advisable, as solutions found by the constant-time model are very likely to be infeasible for the non-linear charging case. Hence, it is important to actually model non-linear charging as such.
- The greater the vehicles' battery capacity, the fewer vehicles are required. However, this becomes marginal once a certain threshold in battery capacity is reached, which is between 90% and 100% of the regular capacity in our experiments. Furthermore, the effect is more severe for non-linear battery charging.
- The required fleet size is smallest for an intermediate number of warehouses and depots. The effect is quite marginal and actual warehouse locations are likely to be much more important, however.

Future research may aim to extend our experiments on the deployment of warehouses and depots in in-plant logistics. While we only considered the effects of the number of warehouses, it stands to reason that their location has a much more severe influence, where there is a multitude of ways depots and warehouses could be arranged in practice. However, such a study would ideally include a case study on typical facility layouts, which is out of the scope of this paper.

While our solution procedure is suitable to solve instances of realistic size (with $n=200$, an instance covers roughly an eight-hour workday), larger instances with $n \geq 1000$ could not be solved when non-linear plug-in charging was assumed. Therefore, future research may develop heuristic approaches inspired by the idea of the branch-and-check procedure. A straightforward way would be to turn the exact master problem into a heuristic beam search procedure. Alternatively, the master problem could be replaced by a meta-heuristic search approach such as tabu search or simulated annealing.

Finally, future research could consider adapting our general solution approach for electric vehicle problems with other or extended objectives as summarized in Table 1. While Li (2014), van Kooten Niekerk et al. (2017), and Adler and Mirchandani (2017) found that column generation performs well as an exact solution approach in that matter, the generally good performance of the BCH procedure in this paper suggests row generation may be a worthwhile consideration, too.

Appendix A.

A1. Determining the parameters $\bar{c}^{\max}_{j,j'}$ and $\bar{c}^{\min}_{j,j'}$

Determining the parameter $\bar{c}^{\max}_{j,j'}$ is not straightforward from its definition in Section 5.1, because we do not know which value of c maximizes the term $\zeta^*(j,j,c)-c$. The same is the case for $\bar{c}^{\min}_{j,j'}$. However, we can reformulate $\bar{c}^{\max}_{j,j'} = \max_{c\in\tilde{C}_j}\{\max_{j''\in S\cup\{0\}}\{\zeta(j,j',j'',c)-c\}\} = \max_{j''\in S\cup\{0\}}\{\max_{c\in\tilde{C}_j}\{\zeta(j,j',j'',c)-c\}\}$. Likewise, we can reformulate $\bar{c}^{\min}_{j,j'} = \min\{c\in\tilde{C}_j \mid \max_{j''\in S\cup\{0\}}\{\zeta(j,j',j'',c)\} \geq \min\{\tilde{C}_{j'}\}\} = \min_{j''\in S\cup\{0\}}\{\min\{c\in\tilde{C}_j \mid \zeta(j,j',j'',c)\geq\min\{\tilde{C}_{j'}\}\}\}$. From the reformulations, it becomes evident that we need to consider every $j''\in S\cup\{0\}$ separately, for which we then calculate

$$\bar{c}^{\max,\,\mathrm{red}}_{j,j'}(j'') = \max_{c\in\tilde{C}_j}\left\{\zeta(j,j',j'',c)-c\right\} \tag{35}$$

and

$$\bar{c}^{\text{min, red}}_{j,j'}(j'') = \min\left\{c \in \tilde{C}_j \mid \zeta(j, j', j'', c) \geq \min\left\{\tilde{C}_{j'}\right\}\right\}, \tag{36}$$

respectively. Moreover, note that we do not need to determine $\bar{c}^{\max}_{j,j'}$ and $\bar{c}^{\min}_{j,j'}$ if $(j, j') \in I$ (cf., Constraints (23) to (25)).

For $j'' = 0$ and $(j, j') \notin I$, it follows that $\bar{c}^{\text{max, red}}_{j,j'}(j'') = -\bar{c}_{j,j'} - \tilde{c}_{j'}$ and $\bar{c}^{\text{min, red}}_{j,j'}(j'') = \min\left\{c \in \tilde{C}_j \mid \ c - \bar{c}_{j,j'} - \tilde{c}_{j'} \geq \min\left\{\tilde{C}_{j'}\right\}\right\} = \min\left\{\tilde{C}_{j'}\right\} + \bar{c}_{j,j'} + \tilde{c}_j$.

For $j'' \in S$, we take the following observation into account. For all charging functions η we consider in this paper, charging is faster (or at least not slower) the lower the SOC. Therefore, the maximum increase in the SOC between j and j' occurs if station $j'' \in S$ is visited with the minimum possible SOC that allows for a feasible execution. Since $\bar{c}^{\text{min, red}}_{j,j'}(j'')$ is the minimum SOC that must be left at j to execute the transition to j' via the deadheading route j'' feasibly, it follows that $\bar{c}^{\text{max, red}}_{j,j'}(j'') = \zeta(j, j', j'', \bar{c}^{\text{min, red}}_{j,j'}(j'')) - \bar{c}^{\text{min, red}}_{j,j'}(j'')$. Hence, it follows that $\bar{c}^{\text{min, red}}_{j,j'}(j'') = \min\{c \in \tilde{C}_j \mid \eta(\eta^{-1}(c - \bar{c}_{j,j''}) + \bar{\tau}(j, j', j'', c)) - \bar{c}_{j'',j'} - \tilde{c}_{j'} \geq \min\{\tilde{C}_{j'}\}\} = \eta(\eta^{-1}(\min\{\tilde{C}_{j'}\} + \bar{c}_{j'',j'} + \tilde{c}_{j'}) - \bar{\tau}(j, j', j'', c)) + \bar{c}_{j,j''}$. Inserting the respective value for c into Expression (35) yields $\bar{c}^{\text{max, red}}_{j,j'}(j'')$ accordingly.

To foster the understanding, the procedure to determine $\bar{c}^{\max}_{j,j'}$ and $\bar{c}^{\min}_{j,j'}$ simultaneously is summarized as pseudo-code in Algorithm 1.

Algorithm 1: Algorithm for determining $\bar{c}^{\max}_{j,j'}$ and $\bar{c}^{\min}_{j,j'}$ simultaneously.

Input: $j, j', \tilde{C}_j, \tilde{C}_{j'}$

$\bar{c}^{\max}_{j,j'} = -\bar{c}_{j,j'} - \tilde{c}_{j'}$; `// maximal increase in the SOC between` j `and` j'`; the initial value results from the case that` $j'' = 0$

$\bar{c}^{\min}_{j,j'} = \bar{c}_{j,j'} + \tilde{c}_{j'} + \min\left\{\tilde{C}_j\right\}$; `// minimum required SOC that must remain at` j `if` j' `should be reached consecutively; the initial value results from the case that` $j'' = 0$

for $j'' \in S$ **do**

  $c := \eta\left(\eta^{-1}\left(\min\{\tilde{C}_{j'}\} + \bar{c}_{j'',j'} - \tilde{c}_{j'}\right) - s_{j'} + e_j + t_{j,j''} + t_{j'',j'} + r\right) + \bar{c}_{j,j''}$;

  $\bar{c}^{\max}_{j,j'} := \max\left\{\zeta(j, j', j'', c) - c, \bar{c}^{\,max}_{j,j'}\right\}$;

  $\bar{c}^{\min}_{j,j'} := \min\left\{c, \bar{c}^{\min}_{j,j'}\right\}$;

Output: $\bar{c}^{\max}_{j,j'}, \bar{c}^{\min}_{j,j'}$

References

Adler, J. D., & Mirchandani, P. B. (2017). The vehicle scheduling problem for fleets with alternative-fuel vehicles. *Transportation Science, 51*(2), 441–456.

Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources, 241*, 680–689.

Beck, J. C. (2010). Checking-up on branch-and-check. In *International conference on principles and practice of constraint programming* (pp. 84–98). Springer.

Bertossi, A. A., Carraresi, P., & Gallo, G. (1987). On some matching problems arising in vehicle scheduling models. *Networks, 17*(3), 271–281. https://doi.org/10.1002/net.3230170303.

Boysen, N., Emde, S., Hoeck, M., & Kauderer, M. (2015). Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research, 242*(1), 107–120.

Briand, C., He, Y., & Ngueveu, S. (2018). Energy-efficient planning for supplying assembly lines with vehicles. *EURO Journal on Transportation and Logistics*, 1–28.

Bunte, S., & Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport, 1*(4), 299–317.

Chao, Z., & Xiaohong, C. (2013). Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. *Procedia-Social and Behavioral Sciences, 96*, 2725–2736.

Codato, G., & Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research, 54*(4), 756–766.

Den Boer, E., Aarnink, S., Kleiner, F., & Pagenkopf, J. (2013). Zero emissions trucks. an overview of state-of-the-art technologies and their potential. https://cedelft.eu/wp-content/uploads/sites/2/2021/04/CE_Delft_4841_Zero_emissions_trucks_Def.pdf. Accessed: 2021-05-12.

Emde, S., Abedinnia, H., & Glock, C. H. (2018). Scheduling electric vehicles making milk-runs for just-in-time delivery. *IISE Transactions, 50*(11), 1013–1025.

Emde, S., & Boysen, N. (2012). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *European Journal of Operational Research, 217*(2), 287–299.

Emde, S., Fliedner, M., & Boysen, N. (2012). Optimally loading tow trains for just-in-time supply of mixed-model assembly lines. *IIE Transactions, 44*(2), 121–135.

Emde, S., & Gendreau, M. (2017). Scheduling in-house transport vehicles to feed parts to automotive assembly lines. *European Journal of Operational Research, 260*(1), 255–267.

Emde, S., & Schneider, M. (2018). Just-in-time vehicle routing for in-house part feeding to assembly lines. *Transportation Science, 52*(3), 657–672.

Emde, S., & Zehtabian, S. (2019). Scheduling direct deliveries with time windows to minimise truck fleet size and customer waiting times. *International Journal of Production Research, 57*(5), 1315–1330.

Fathi, M., Rodríguez, V., & Alvarez, M. J. (2014). A novel memetic ant colony optimization-based heuristic algorithm for solving the assembly line part feeding problem. *The International Journal of Advanced Manufacturing Technology, 75*(1–4), 629–643.

Fathi, M., Rodríguez, V., Fontes, D. B., & Alvarez, M. J. (2016). A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines. *International Journal of Production Research, 54*(3), 878–893.

Freling, R., & Paixão, J. M. P. (1995). Vehicle scheduling with time constraint. In *Computer-aided transit scheduling* (pp. 130–144). Springer.

Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research, 245*(1), 81–99.

Golz, J., Gujjula, R., Günther, H.-O., Rinderer, S., & Ziegler, M. (2012). Part feeding at high-variant mixed-model assembly lines. *Flexible Services and Manufacturing Journal, 24*(2), 119–141.

Gschwind, T., Irnich, S., Tilk, C., & Emde, S. (2019). Branch-cut-and-price for scheduling deliveries with time windows in a direct shipping network. *Journal of Scheduling*, 1–15.

Haghani, A., & Banihashemi, M. (2002). Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. *Transportation Research Part A: Policy and Practice, 36*(4), 309–333.

He, Y., Song, Z., & Liu, Z. (2019). Fast-charging station deployment for battery electric bus systems considering electricity demand charges. *Sustainable Cities and Society, 48*, 101530.

Hooker, J. (2011). *Logic-based methods for optimization: Combining optimization and constraint satisfaction*: vol. 2. John Wiley & Sons.

Hu, L., Zhou, B., & Li, Y. (2017). An energy saving scheduling method for just in time material handling in mixed-model assembly line. In *ASME 2017 12th international manufacturing science and engineering conference collocated with the JSME/ASME 2017 6th international conference on materials and processing*. American Society of Mechanical Engineers. https://doi.org/10.1115/MSEC2017-2985.

Ibarra-Rojas, O., Delgado, F., Giesen, R., & Muñoz, J. (2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological, 77*, 38–75.

Janovec, M., & Koháni, M. (2019a). Battery degradation impact on the electric bus fleet scheduling. In *2019 International conference on information and digital technologies (IDT)* (pp. 190–197). IEEE.

Janovec, M., & Koháni, M. (2019b). Exact approach to the electric bus fleet scheduling. *Transportation Research Procedia, 40*, 1380–1387.

Kisacikoglu, M. C., Ozpineci, B., & Tolbert, L. M. (2011). Reactive power operation analysis of a single-phase EV/PHEV bidirectional battery charger. In *8th International conference on power electronics-ECCE Asia* (pp. 585–592). IEEE.

Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H., & Spieksma, F. C. (2007). Interval scheduling: A survey. *Naval Research Logistics (NRL), 54*(5), 530–543.

Kovalyov, M. Y., Ng, C., & Cheng, T. E. (2007). Fixed interval scheduling: Models, applications, computational complexity and algorithms. *European Journal of Operational Research, 178*(2), 331–342.

Li, J.-Q. (2014). Transit bus scheduling with limited energy. *Transportation Science, 48*(4), 521–539.

Lieb, C., Klenk, D.-W.-I. E., Galka, D.-W.-I. S., & Keuntje, C. (2017). Einsatz von routenzugsystemen zur produktionsversorgung. *Studie zu Planung, Steuerung und Betrieb. Lehrstuhl für Fördertechnik Materialfluss Logistik. München*.

Lin, Y., Zhang, K., Shen, Z.-J. M., Ye, B., & Miao, L. (2019). Multistage large-scale charging station planning for electric buses considering transportation network and power grid. *Transportation Research Part C: Emerging Technologies, 107*, 423–443.

Liu, W. (2013). *Introduction to hybrid vehicle system modeling and control*. John Wiley & Sons.

Liu, Z., Song, Z., & He, Y. (2018). Planning of fast-charging stations for a battery electric bus system under energy consumption uncertainty. *Transportation Research Record, 2672*(8), 96–107.

Messaoudi, B., & Oulamara, A. (2019). Electric bus scheduling and optimal charging. In C. Paternina-Arboleda, & S. Voß (Eds.), *Computational logistics* (pp. 233–247). Cham: Springer International Publishing.

Olsen, N., & Kliewer, N. (2018). Electric vehicle scheduling–a study on charging modeling for electric vehicles. In *Operations research proceedings 2016* (pp. 653–658). Springer.

Olsen, N., & Kliewer, N. (2020). Scheduling electric buses in public transport: Modeling of the charging process and analysis of assumptions. *Logistics Research, 13*(1), 4.

Pelletier, S., Jabali, O., Laporte, G., & Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transportation Research Part B: Methodological, 103*, 158–187.

Reuer, J., Kliewer, N., & Wolbeck, L. (2015). The electric vehicle scheduling problem: A study on time-space network based and heuristic solution approaches. In *Proceedings of the 13th conference on advanced systems in public transport (CASPT), Rotterdam*.

Saha, J. (1970). An algorithm for bus scheduling problems. *Journal of the Operational Research Society, 21*(4), 463–474.

Sassi, O., & Oulamara, A. (2017). Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches. *International Journal of Production Research, 55*(2), 519–535.

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

Still (2019). Ltx technical data. https://data.still.de/assets/products/Vehicles/Platform_Trucks_and_Tractors/LTX_50-T06/pdf/LTX_20_LTX_50_EN_TD_web.pdf?mod=1530628525&download=1&s=54a8343b5761ed631c8427172f20b6fe. Accessed: 2020-05-07.

Sweda, T. M., Dolinskaya, I. S., & Klabjan, D. (2016). Optimal recharging policies for electric vehicles. *Transportation Science, 51*(2), 457–479.

Teng, J., Chen, T., & Fan, W. D. (2020). Integrated approach to vehicle scheduling and bus timetabling for an electric bus line. *Journal of Transportation Engineering, Part A: Systems, 146*(2), 04019073.

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In *International conference on principles and practice of constraint programming* (pp. 16–30). Springer.

van Kooten Niekerk, M. E., van den Akker, J., & Hoogeveen, J. (2017). Scheduling electric vehicles. *Public Transport, 9*(1–2), 155–176.

Vetter, J., Novák, P., Wagner, M. R., Veit, C., Möller, K.-C., Besenhard, J., … Hammouche, A. (2005). Ageing mechanisms in lithium-ion batteries. *Journal of power sources, 147*(1–2), 269–281.

Wang, H., & Shen, J. (2007). Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation, 190*(2), 1237–1249.

Wang, Y., Huang, Y., Xu, J., & Barclay, N. (2017). Optimal recharging scheduling for urban electric buses: A case study in davis. *Transportation Research Part E: Logistics and Transportation Review, 100*, 115–132.

Wen, M., Linde, E., Ropke, S., Mirchandani, P., & Larsen, A. (2016). An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research, 76*, 73–83.

Yao, E., Liu, T., Lu, T., & Yang, Y. (2020). Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society, 52*, 101862.

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

Zhou, B., & Peng, T. (2017). Scheduling the in-house logistics distribution for automotive assembly lines with just-in-time principles. *Assembly Automation, 37*(1), 51–63.

Zhou, B.-h., & Tan, F. (2018). Electric vehicle handling routing and battery swap station location optimisation for automotive assembly lines. *International Journal of Computer Integrated Manufacturing, 31*(10), 978–991.

Zhou, G.-J., Xie, D.-F., Zhao, X.-M., & Lu, C. (2020). Collaborative optimization of vehicle and charging scheduling for a bus fleet mixed with electric and traditional buses. *IEEE Access, 8*, 8056–8072.