



Discrete Optimization

Routing a mixed fleet of electric and conventional vehicles

Dominik Goeke^a, Michael Schneider^{b,*}^a Chair of Business Information Systems and Operations Research, University of Kaiserslautern, Germany^b DB Schenker Endowed Assistant Professorship for Logistics Planning and Information Systems, TU Darmstadt, Germany

ARTICLE INFO

Article history:

Received 31 July 2014

Accepted 25 January 2015

Available online 11 February 2015

Keywords:

Electric vehicles

Energy consumption

Vehicle routing

Metaheuristic

Green logistics

ABSTRACT

In this paper, we propose the Electric Vehicle Routing Problem with Time Windows and Mixed Fleet (E-VRPTWMF) to optimize the routing of a mixed fleet of electric commercial vehicles (ECVs) and conventional internal combustion commercial vehicles (ICCVs). Contrary to existing routing models for ECVs, which assume energy consumption to be a linear function of traveled distance, we utilize a realistic energy consumption model that incorporates speed, gradient and cargo load distribution. This is highly relevant in the context of ECVs because energy consumption determines the maximal driving range of ECVs and the recharging times at stations. To address the problem, we develop an Adaptive Large Neighborhood Search algorithm that is enhanced by a local search for intensification. In numerical studies on newly designed E-VRPTWMF test instances, we investigate the effect of considering the actual load distribution on the structure and quality of the generated solutions. Moreover, we study the influence of different objective functions on solution attributes and on the contribution of ECVs to the overall routing costs. Finally, we demonstrate the performance of the developed algorithm on benchmark instances of the related problems VRPTW and E-VRPTW.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In Europe, recent years have seen a steady increase in energy costs while more and more laws are passed to regulate the emission of greenhouse gases in the transportation sector. These external factors and the society's rising environmental and social awareness have triggered numerous green initiatives at companies. In the logistics field, electric commercial vehicles (ECVs) are now considered a serious alternative to conventional internal combustion commercial vehicles (ICCVs). ECVs have no local greenhouse gas emission and produce only minimal noise, however, they are currently hardly competitive with ICCVs from a cost point of view (Davis & Figliozzi, 2013). Nevertheless, several companies have started to employ ECVs in their last-mile delivery operations, e.g., in the field of small-package shipping (Kleindorfer, Neboian, Roset, & Spinler, 2012) or the distribution of food (National Renewable Energy Laboratory, 2014) and beverages (Heineken International, 2014). Moreover, governments and private companies are starting to provide the required infrastructure to further boost this electrification trend (International Energy Agency, 2012, 2013; Tesla Motors, Inc., 2014).

One important aspect to render ECVs more competitive is to consider their special characteristics – a limited driving range and the

potential need to recharge en route – in the planning of last-mile delivery operations employing ECVs. These distribution tasks are generally represented as vehicle routing problems (VRPs), whose goal is to find minimum-cost routes to serve a given set of customers from a central depot (Toth & Vigo, 2014). The first VRPs to address ECVs (or alternative fuel vehicles) with a limited driving range and the possibility of recharging (refueling) at dedicated stations have recently been presented in the literature (Conrad & Figliozzi, 2011; Erdogan & Miller-Hooks, 2012; Schneider, Stenger, & Goeke, 2014). Although two important constraints of last-mile delivery operations, namely vehicle capacity constraints and customer time windows have already been considered (Conrad & Figliozzi, 2011; Schneider, Stenger, & Goeke, 2014), many relevant real-life constraints are not yet covered by routing models for ECVs and dedicated solution methods.

In this work, we consider two important aspects for route planning with ECVs:

Mixed fleet: Most companies do not operate pure ECV fleets but are gradually introducing ECVs into their existing ICCV fleet. Therefore, our route planning method is able to handle a mixed fleet of ECVs and ICCVs. Compared to ICCVs, energy costs for operating ECVs are generally lower while labor costs may increase due to time spent on potentially necessary recharging activities along the routes. Energy and labor costs are among the main components of total operating costs (see, e.g., Bektaş & Laporte, 2011), so

* Corresponding author. Tel.: +4915209818017.

E-mail addresses: dominik.goeke@wiwi.uni-kl.de (D. Goeke), schneider@bwl.tu-darmstadt.de (M. Schneider).

high-quality route planning with a mixed vehicle fleet has to consider the cost tradeoff between the two vehicle types.

Energy consumption: Real-life energy consumption is not a linear function of traveled distance as assumed in the models of Erdogan and Miller-Hooks (2012) and Schneider, Stenger, and Goeke (2014). We use realistic energy consumption functions of ECVs and ICCVs that incorporate vehicle speed, gradients and cargo load. In recent years, realistic energy consumption models have started to play an important role in routing models that consider fuel costs and vehicle emissions (Bektaş & Laporte, 2011; Jabali, Van Woensel, & de Kok, 2012). In the context of ECVs, energy consumption determines electricity costs on the one hand, but, more importantly, also the driving range of an ECV and thus the latest possible moment at which a recharge has to take place in order to prevent an ECV from getting stranded.

We propose the Electric VRP with Time Windows and Mixed Fleet (E-VRPTWMF) to determine optimal routes (according to different alternative objective functions considered in this work) for a given mixed fleet of ECVs and ICCVs. E-VRPTWMF incorporates time window and vehicle capacity constraints. We assume that ECVs can be recharged at any of the available stations causing a recharging time that depends on the battery level on arrival at the station. The energy consumption of ICCVs is calculated by means of the model presented in Demir, Bektaş, and Laporte (2012) and we extend this model to compute the battery energy consumption of ECVs.

As E-VRPTWMF extends the notoriously hard-to-solve VRPTW, exact methods will not be able to solve instances of realistic size within fast computation times. Therefore, we develop a heuristic solution method to address the problem, namely an Adaptive Large Neighborhood Search (ALNS) enhanced by a Local Search (LS) for intensification. Besides new operators considering recharging stations, our ALNS features several new ideas: (i) an adaptive mechanism to choose the number of customers to be removed in each iteration, (ii) the use of surrogate violations in order to handle the complexity of calculating time window and battery capacity violations, and (iii) an acceptance criterion taking into account the different penalty factors that were used when calculating the objective value of the solutions to be compared.

In numerical studies, we assess the performance of our ALNS on benchmark instances of related problems: ALNS achieves convincing results on the well-studied VRPTW benchmark of Solomon (1987), and outperforms previous methods on the E-VRPTW benchmark set of Schneider, Stenger, and Goeke (2014). In addition, we generate a set of new E-VRPTWMF instances based on the Pollution Routing Problem (PRP) benchmark of Demir et al. (2012). In experiments on these newly designed instances, we find that consideration of the actual load strongly improves the quality of the generated solutions in comparison to solutions that are generated based on load estimates. Moreover, we find that a large number of solutions that are generated with “optimistic” load estimates are actually infeasible due to battery capacity or time window violations. We further show that our ALNS works effectively with all of the investigated cost functions and that the traditional objective of minimizing traveled distance fails to produce high-quality solutions if routing costs including energy, labor and battery depreciation are considered. The choice of objective function additionally has a strong influence on the level of usage of the ECVs in the fleet.

This paper is organized as follows: In Section 2, we briefly review the literature related to E-VRPTWMF. Section 3 introduces the energy consumption models for ECVs and ICCVs. Section 4 presents a mixed-integer program of E-VRPTWMF. The ALNS is detailed in Section 5. The parameter setting, the generation of new E-VRPTWMF instances and the numerical studies on the new instances and on the test instances

of related problems are presented in Section 6. Section 7 summarizes and concludes the paper.

2. Literature

In the following, we discuss the literature related to the E-VRPTWMF. First, we review works on routing alternative fuel vehicles. Second, VRP papers that explicitly model energy consumption or account for the impact of load distribution are presented. Third, we discuss other related fields.

Conrad and Figliozzi (2011) study the Recharging VRP, in which vehicles with limited range can recharge at certain customer locations. Time window constraints are considered and a fixed recharging time is assumed. The authors compute bounds to predict average tour lengths and study the impact of driving range, recharging times, and time window existence. Erdogan and Miller-Hooks (2012) propose two heuristics for the Green VRP. In this problem, alternative fuel is only available at dedicated points that have to be visited en route. Refueling time is assumed fixed and no capacity or time window constraints are included. Other works addressing the Green VRP or extensions of this problem are (Felipe, Ortuño, Righini, & Tirado, 2014; Montoya, Guéret, Mendoza, & Villegas, 2014; Schneider, Stenger, & Hof, 2014). Schneider, Stenger, and Goeke (2014) develop a hybrid of Variable Neighborhood Search (VNS) and Tabu Search (TS) to address E-VRPTW, in which ECVs with a limited battery capacity may visit recharging stations en route, and customer time windows and vehicle capacities have to be respected. Recharging time is proportional to the amount of energy required to recharge the battery to full capacity. Desaulniers, Errico, Irnich, and Schneider (2014) present branch-price-and-cut algorithms to address four variants of the E-VRPTW. In a recent working paper, Hiemann, Puchinger, and Hartl (2014) combine the E-VRPTW and the Fleet Size and Mix VRP with Fixed costs (FSMF). In the resulting E-FSMVRPTW an unlimited number of ECVs with different battery capacities and vehicle-independent routing costs are available. Cost-optimal routes are determined by means of an ALNS enhanced by a labeling algorithm. Barco, Guerra, Muñoz, and Quijano (2013) propose a comprehensive approach for planning the deployment of electric vehicles in an airport shuttle service. They first determine a minimal consumption graph, on which routing decisions under consideration of a limited battery capacity are made. The assignment of vehicles to routes and the scheduling of recharges is determined by an evolutionary algorithm. Finally, Preis, Frank, and Nachtigall (2014) investigate an ECV routing model with customer time windows, fixed recharging times, and the goal of minimizing total energy consumption, which depends on gradients and cargo load. A simple TS algorithm based on a relocate operator is presented.

The second strand of relevant literature integrates energy considerations and the resulting fuel consumption and emissions of ICCVs into routing models. Bektaş and Laporte (2011) propose the PRP, in which they estimate the price of pollution and introduce it as part of the objective function, besides costs for driver wages and fuel consumption. They allow the choice between different speed levels for arcs and consider speed, gradients, and load to calculate the fuel consumption and corresponding emissions. Demir, Bektaş, and Laporte (2011) provide a comparison of vehicle emission models. Demir et al. (2012) propose an ALNS and a speed optimization algorithm for the PRP. Computational experiments show moderate run-times for problem sizes with 200 customers. Demir, Bektaş, and Laporte (2014a) address a bi-objective function for the PRP that models the conflicting targets of minimizing driver time and usage of fuel and thus avoid the problematic representation of emissions in terms of monetary cost. An extensive review of related literature can be found in Demir, Bektaş, and Laporte (2014b). Kopfer, Schönberger, and Kopfer (2014) use CPLEX to solve a heterogeneous VRP with the objective of minimizing fuel consumption. To this end, the authors derive linear relationships

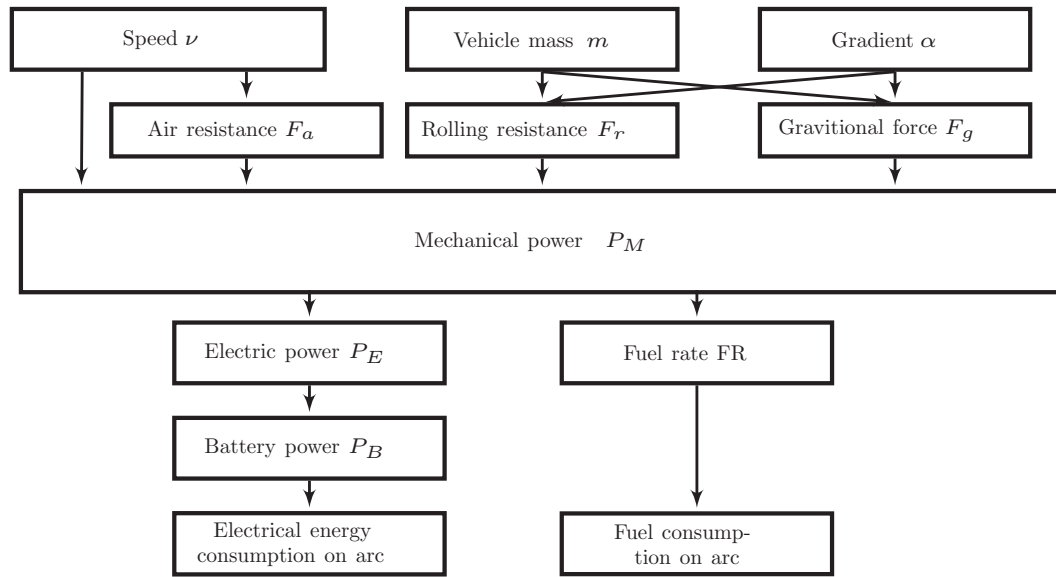


Fig. 1. Calculation of required energy and fuel.

between fuel consumption and cargo load for the different vehicle types.

Xiao, Zhao, Kaku, and Xu (2012) calculate a fuel consumption rate depending on the mass of the remaining cargo and they find significant fuel saving potential for the CVRP. Zhang, Tang, and Fung (2011) propose a multi-depot VRP that includes, in the objective function, a cost term that increases linearly with the amount of cargo and the traveled distance and which depends on an empirical cost factor. Another related problem is the fuel or emission shortest-path problem, which may be used to define the arc set of a VRP. Nie and Li (2013) propose such a model to minimize total operating cost while meeting a limit on emissions.

E-VRPTWMF is further related to VRPs with intermediate replenishment facilities (Crevier, Cordeau, & Laporte, 2007; Hemmelmayer, Doerner, Hartl, & Rath, 2013) or with distance constraints (Laporte, Nobert, & Desrochers, 1985) due to the similar structure of the problems, to refueling problems with a dense infrastructure (Bousonville, Hartmann, Melo, & Kopfer, 2011; Suzuki, 2012), to energy shortest path problems (Artmeier, Haselmayer, Leucker, & Sachenbacher, 2010), and in a more general sense to the field of green logistics from an OR perspective (Dekker, Bloemhof, & Mallidis, 2012; Sbihi & Eglese, 2010). Moreover, the planning of electric infrastructure has attracted increasing research interest (He, Wu, Yin, & Guan, 2013; Mak, Rong, & Shen, 2013; Nie & Ghamami, 2013; Wang & Lin, 2013). Finally, Davis and Figliozzi (2013) and Feng and Figliozzi (2013) address the competitiveness of ECVs. Davis and Figliozzi (2013) show that ECVs are competitive in comparison to ICCVs for certain application scenarios with large route distances and low vehicle speeds.

3. Energy consumption models

Our VRP model incorporates two specific characteristics of employing ECVs, namely a reduced operating range and the possibility to recharge at certain stations in order to increase this range. The necessity to visit a recharging station and the recharging time at the station depend on the battery level of the ECV, which itself depends on the energy consumption along the route. Previous works on vehicle routing problems with refueling or recharging (Erdogan & Miller-Hooks, 2012; Schneider, Stenger, & Goeke, 2014) assume energy consumption to be a linear function of the traveled distance. In this work, we use a more realistic model and consider the following factors that influence the energy consumption of a vehicle:

Vehicle mass: The mass of the vehicle is composed of its curb mass and the load. Concerning the routing of the vehicle, the order in which customers are visited and cargo is unloaded, i.e., the distribution of the load during the course of a route, can strongly influence the energy consumption on the route.

Speed: Accelerating a vehicle and keeping the vehicle in motion at a certain speed consumes energy due to the aerodynamic and rolling resistance that has to be overcome. We assume travel speeds to be constant on each arc, i.e., we neglect acceleration phases, but vehicle speed may be different for different arcs. Note that in principle vehicle speed could be handled as a decision variable and could be increased in order to fulfill time window requirements and be reduced to decrease energy consumption (see, e.g., Demir et al., 2014b). As traffic conditions have a strong impact on vehicle speed, we chose to refrain from this modeling.

Gradient of the terrain: We assume a non-flat terrain with grades. Going uphill requires higher amounts of energy than traveling flat terrain or going downhill. ECVs may be able to recharge their battery when going downhill, a process called recuperation.

In the following, we present the methods for determining the energy consumption of ECVs (Section 3.1) and ICCVs (Section 3.2).

3.1. Energy consumption of electric vehicles

We calculate the energy consumption of an ECV in three steps as shown in Fig. 1. First, we determine the mechanical power P_M using the model presented in Bektaş and Laporte (2011), which determines energy consumption based on the factors described above (mass, speed, gradient) and the physical environment (road surface, vehicle dimensions, engine properties). Second, P_M is translated into the electric power P_E that the electric motor needs to provide the required amount of mechanical power. This amount is determined by the efficiency of the electric motor.

Third, the required electric power is converted into the amount of power that has to be taken from the battery (P_B), which depends on the battery efficiency. In the following, these three steps are described in more detail.

The mechanical power P_M is needed to overcome rolling resistance and aerodynamic resistance and is influenced by the gravitational

force. Let m denote the total vehicle mass, g the gravitational constant, c_r the rolling friction coefficient (which depends on tire pressure, road surface conditions and other factors) and α the gradient angle. Then the rolling resistance F_r can be determined as

$$F_r = c_r \cdot m \cdot g \cdot \cos(\alpha).$$

With v denoting the speed, c_d the aerodynamic drag coefficient, ρ_a the air density and A_f the frontal area of the vehicle, the aerodynamic resistance is:

$$F_a = \frac{1}{2} \cdot \rho_a \cdot A_f \cdot c_d \cdot v^2.$$

Adding the gravitational force $F_g = m \cdot g \cdot \sin(\alpha)$, the total mechanical power P_M is:

$$P_M = \left(m \cdot a + \frac{1}{2} \cdot c_d \cdot \rho \cdot A \cdot v^2 + m \cdot g \cdot \sin(\alpha) + c_r \cdot m \cdot g \cdot \cos(\alpha) \right) \cdot v, \quad (1)$$

where a denotes the acceleration. As mentioned above, a is assumed to be zero in the following as we do not consider acceleration and braking processes.

In the second step, the electric power P_E that is needed to achieve the required mechanical power P_M is calculated, taking into account energy losses that occur in the electric engine. To this end, we use the Quantized State System (QSS) model of an electric engine presented in (Guzzella & Amstutz, 2005). The authors provide values for the engine efficiency at given pairs of torque and rotational velocity. We directly convert torque and rotational velocity to mechanical power. Using homogeneous regression, i.e. a linear regression with a y-axis intercept of zero, on the converted values, we obtain the following relationship between P_M and the discharged electric energy P_E^d (respectively the recuperated electric energy P_E^r) that is described by the regression coefficient ϕ :

$$\begin{aligned} P_E^d &= \phi^d \cdot P_M \text{ for } 0 \text{ kilowatts} \leq P_M \leq 100 \text{ kilowatts} \\ P_E^r &= \phi^r \cdot P_M \text{ for } -100 \text{ kilowatts} \leq P_M < 0 \text{ kilowatts.} \end{aligned}$$

The determined relationship is adequate for a mechanical power demand of up to 100 kilowatts, which we consider a realistic threshold for commonly employed ECV models.

In the third step, we compute the battery power P_B that is necessary to obtain the required electric energy P_E . For this purpose, we use the relationship for converting electric power P_E into battery power P_B for a lithium-ion accumulator presented in van Keulen, de Jager, Serrarens, and Steinbuch (2010). The presented relationship does not account for external effects (e.g., the ambient temperature) or the influence of the current charge level of the battery. We approximate the given relation by means of a homogeneous regression and obtain as final equations for discharging (P_B^d) and recuperating (P_B^r)

$$\begin{aligned} P_B^d &= \varphi^d \cdot P_E \text{ for } P_E \geq 0 \text{ kilowatts} \\ P_B^r &= \varphi^r \cdot P_E \text{ for } P_E < 0 \text{ kilowatts,} \end{aligned}$$

with the regression coefficient φ that describes the battery efficiency.

To compute energy consumption on a graph, vehicle mass is defined as a function of the currently loaded amount of cargo u , i.e., $m(u) = m_c + m_u \cdot u$, where m_c is the curb mass of the vehicle and m_u the mass of one unit of cargo. Let u_j denote the amount of cargo when arriving at customer j (before unloading), i.e., the amount required to fulfill the demand at customer j and all following customers. Further, let $P_{ij}(u_j)$ denote the constant mechanical power demand over the course of an arc (i, j) with a load that is equivalent to the remaining customer demand:

$$P_{ij}(u_j) = \left(\frac{1}{2} \cdot c_d \cdot \rho \cdot A \cdot v^2 + m(u_j) \cdot g \cdot (\sin(\alpha_{ij}) + c_r \cdot \cos(\alpha_{ij})) \right) \cdot v_{ij}$$

Then, the battery power demand on this arc is given by $P_B(P_E(P_{ij}(u_j)))$. Multiplying with the associated travel time t_{ij} of the arc yields the electric energy consumption b_{ij} for traveling this particular arc:

$$b_{ij}(u_j) = \begin{cases} \phi^d \cdot \varphi^d \cdot P_{ij}(u_j) \cdot t_{ij} & \text{if } P_{ij}(u_j) \geq 0 \text{ kilowatts} \\ \phi^r \cdot \varphi^r \cdot P_{ij}(u_j) \cdot t_{ij} & \text{if } P_{ij}(u_j) < 0 \text{ kilowatts.} \end{cases} \quad (2)$$

All variables except the amount of cargo u_j are constant for a given arc. Therefore, it is possible to split $b_{ij}(u_j)$ into a fixed and a load-dependent part and precompute parts of the expression.

3.2. Energy consumption of combustion engines

Mechanical power can be directly translated to diesel fuel consumption as shown in Demir et al. (2012). The fuel rate is given by $FR = \frac{\xi}{\kappa \cdot \psi} (kND + \frac{P_M}{\eta \cdot \eta_{tf}})$, where P_M is the mechanical power as defined in Eq. (1), ξ the fuel-to-air mass ratio, κ the heating value of typical diesel fuel, k the engine friction factor, N the engine speed, D the engine displacement, ψ a factor converting the fuel rate from grams per second to liters per second, η the efficiency parameter for diesel engines and η_{tf} the drive train efficiency. We assume that driving downhill in gear can reduce the engine consumption down to 0. The constant fuel rate over the course of an arc is given by:

$$FR_{ij}(u_j) = \max \left(\frac{\xi}{\kappa \cdot \psi} \left(kND + \frac{P_{ij}(u_j)}{\eta \cdot \eta_{tf}} \right), 0 \right).$$

Thus, the fuel consumption of an ICCV traversing an arc (i, j) with cargo u_j can be calculated as:

$$f_{ij}(u_j) = t_{ij} \cdot FR_{ij}(u_j). \quad (3)$$

4. The Electric Vehicle Routing Problem with Time Windows and Mixed Fleet (E-VRPTWMF)

We define E-VRPTWMF on a complete, directed graph $G = (V'_{0,N+1}, A)$. $V = \{1, \dots, N\}$ denotes the set of customers, F the set of recharging stations, F' the set of visits to vertices in F and V' denotes the union of V and F' . Vertices 0 and $N+1$ denote instances of the depot, and all routes start at 0 and end at $N+1$. Inclusion of one or both depot instances is indicated by subscripting the respective set, e.g., the set $V'_{0,N+1}$ contains both instances of the depot. The set of arcs is given by $A = \{(i, j) \mid i, j \in V'_{0,N+1}, i \neq j\}$.

With each vertex $i \in V'_{0,N+1}$, we associate a nonnegative demand q_i ($q_i = 0$, $i \notin V$), a nonnegative service time s_i ($s_i = 0$, $i \notin V$) and a time window $[e_i, l_i]$, within which service at the customer has to start. Time windows are hard, i.e., starting service late is not allowed but waiting in case of early arrival is possible. The time window $[e_0, l_{N+1}]$ at the depot specifies the scheduling horizon of the problem. Each arc is described by the following properties: distance d_{ij} , travel speed v_{ij} , travel time $t_{ij} = d_{ij}/v_{ij}$ and a factor α_{ij} representing the gradient of the terrain.

A mixed vehicle fleet of fixed size consisting of m_E ECVs and m_{IC} ICCVs is positioned at the depot. All vehicles have the same maximal cargo loading capacity Q . On each visit to a recharging station, ECVs are recharged to their maximum battery capacity B . The recharging time depends on the recharging rate r and the difference between B and charge level on arrival at the station. Electricity and fuel consumption of a vehicle traveling an arc are determined by the models introduced in Section 3.

Thus, E-VRPTWMF can be formulated as a (nonlinear) mixed-integer program. For every arc $(i, j) \in A$, we define the following binary decision variables: Variables x_{ij}^E are equal to 1 if an arc (i, j) is traveled by an ECV, variables x_{ij}^{IC} if the arc is traveled by an ICCV. Otherwise, the decision variables are 0. Variables τ_i define the arrival time, u_j the remaining cargo (before unloading) and y_i the remaining charge level on arrival at vertex $i \in V'_{0,N+1}$. Table 1 summarizes the variables and parameters of our model.

Table 1
Variable and parameter definitions of the E-VRPTWMF model.

$0, N+1$	Depot instances
F'	Set of visits to recharging stations, dummy vertices of the set of recharging stations F
V	Set of customer vertices $V = \{1, \dots, N\}$
V'	Set of customers and recharging visits: $V' = V \cup F'$
d_{ij}	Distance between vertices i and j
t_{ij}	Travel time between vertices i and j
Q	Vehicle capacity
r	Recharging rate
B	Battery capacity
$b_{ij}(u_j)$	Electric energy consumption between between vertices i and j depending on cargo u_j (Eq. (2))
$f_{ij}(u_j)$	Fuel consumption between between vertices i and j depending on cargo u_j (Eq. (3))
q_i	Demand of vertex i , 0 if $i \notin V$
e_i	Earliest start of service at vertex i
l_i	Latest start of service at vertex i
s_i	Service time at vertex i ($s_0, s_{N+1} = 0$)
τ_i	Decision variable specifying the time of arrival at vertex i
u_i	Decision variable specifying the remaining cargo on arrival at vertex i
y_i	Decision variable specifying the remaining battery capacity on arrival at vertex i
x_{ij}^E	Binary decision variable indicating if arc (i, j) is traveled by an ECV
x_{ij}^C	Binary decision variable indicating if arc (i, j) is traveled by an ICCV

In this work, we consider three alternative objective functions for E-VRPTWMF:

1. Minimize traveled distance f_d :

$$f_d = \sum_{i \in V_0, j \in V'_{N+1}, i \neq j} d_{ij} (x_{ij}^E + x_{ij}^C) \rightarrow \min \quad (4)$$

2. Minimize cost f_c , consisting of the cost for vehicle propulsion and labor cost. Cost factors c^E and c^C denote the cost of electric energy, respective diesel fuel and c^D the driver wage per time unit:

$$f_c = \sum_{i \in V_0, j \in V'_{N+1}, i \neq j} [c^E \cdot b_{ij}(u_j) \cdot x_{ij}^E + c^C \cdot f_{ij}(u_j) \cdot x_{ij}^C] + \sum_{i \in V'} c^D (x_{iN+1}^E + x_{iN+1}^C) (\tau_i + s_i + t_{iN+1}) \rightarrow \min \quad (5)$$

3. Minimize cost including cost for battery replacement f_{cb} . The distance traveled by ECVs is multiplied by a cost factor c^B that accounts for the cost of battery replacement. It is expressed as cost per kilometer by dividing acquisition cost by the expected total lifetime mileage.

$$f_{cb} = f_c + \sum_{i \in V_0, j \in V'_{N+1}, i \neq j} c^B \cdot d_{ij} \cdot x_{ij}^E \rightarrow \min \quad (6)$$

The constraints of E-VRPTWMF are as follows:

$$\sum_{j \in V'} (x_{ij}^E + x_{ij}^C) = 1 \quad \forall i \in V \quad (7)$$

$$\sum_{j \in V_{N+1}, i \neq j} x_{ij}^E \leq 1 \quad \forall i \in F' \quad (8)$$

$$\sum_{j \in V_{N+1}, i \neq j} x_{ij}^E - \sum_{j \in V_0, i \neq j} x_{ji}^E = 0 \quad \forall i \in V' \quad (9)$$

$$\sum_{j \in V_{N+1}, i \neq j} x_{ij}^C - \sum_{j \in V_0, i \neq j} x_{ji}^C = 0 \quad \forall i \in V \quad (10)$$

$$\sum_{j \in V'} x_{0j}^E \leq m_E \quad (11)$$

$$\sum_{j \in V'} x_{0j}^C \leq m_C \quad (12)$$

$$\tau_i + (s_i + t_{ij}) (x_{ij}^E + x_{ij}^C) - l_0 (1 - x_{ij}^E - x_{ij}^C) \leq \tau_j \quad \forall i \in V_0, \forall j \in V'_{N+1}, i \neq j \quad (13)$$

$$\tau_i + t_{ij} \cdot x_{ij}^E + r (B - y_i) x_{ij}^E - (l_0 + rB) (1 - x_{ij}^E) \leq \tau_j \quad \forall i \in F', \forall j \in V'_{N+1}, i \neq j \quad (14)$$

$$e_i \leq \tau_i \leq l_i \quad \forall i \in V'_{0,N+1} \quad (15)$$

$$u_i - q_i (x_{ij}^E + x_{ij}^C) + Q (1 - x_{ij}^E - x_{ij}^C) \geq u_j \geq 0 \quad \forall i \in V_0, \forall j \in V'_{N+1}, i \neq j \quad (16)$$

$$0 \leq u_0 \leq Q \quad (17)$$

$$y_i - b_{ij}(u_j) \cdot x_{ij}^E + B (1 - x_{ij}^E) \geq y_j \quad \forall i \in V, \forall j \in V'_{N+1}, i \neq j \quad (18)$$

$$B - b_{ij}(u_j) \cdot x_{ij}^E \geq y_j \quad \forall i \in F_0', \forall j \in V'_{N+1}, i \neq j \quad (19)$$

$$B \geq y_i \geq 0 \quad \forall i \in V'_{0,N+1} \quad (20)$$

$$x_{ij}^E, x_{ij}^C \in \{0, 1\} \quad \forall i, j \in V'_{0,N+1}, i \neq j \quad (21)$$

Constraints (7) enforce that each customer visit has exactly one successor. Constraints (8) guarantee that each visit to a recharging station is used at most once, i.e., not all recharging visit vertices must be used. Flow conservation constraints (9) and (10) guarantee for each vertex that the number of incoming arcs is equal to the number of outgoing arcs for each vehicle type. Constraints (11) and (12) ensure that the maximum number of employed vehicles adheres to the fleet composition. Constraints (13) and (14) link arrival times at vertices i and j if the arc from i to j is traveled. Constraints (14) covers the case with recharging visits: Here, recharging times that depend on the remaining charge level y_i when arriving at station i have to be taken into account. Constraints (15) guarantee arrival within the time window at each vertex. Constraints (16) enforce the fulfillment of demand at customer vertices. Constraint (17) restricts the initial cargo load level to the maximum capacity of a vehicle.

Constraints (18) set the battery level at a vertex succeeding a customer visit in accordance with the energy consumption for discharging or recuperating. For vertices succeeding a visit to a recharging station or the depot, Constraints (19) set the battery level equal to the maximum battery capacity reduced by the energy required on the respective arc. Here, the battery is either discharged or recuperation takes place. Constraints (20) restricts the charge level to the maximum battery capacity in order to prevent recuperation beyond the maximal capacity. Finally, binary decision variables are defined in constraints (21).

Although our model covers several real-world aspects of ECVs, we still use some simplifications. Gradient and speed are considered fixed over the course of an arc. However, a more fine-grained topology and acceleration patterns can be integrated, e.g., by considering a path $p(i, j)$ instead of the arc (i, j) . A path $p(i, j)$ is defined by a fixed sequence

```

preprocessArcList()
S ← generateInitialSolution(mE, mIC)
while termination criterion not met do
    {randomly draw the number of customers δ to remove from different intervals. Probabilities of intervals
     are given by π||}
    δ ← drawNumberOfCustomersToRemove(π||)
    {Apply randomly selected destroy operator. Probabilities of destroy operators are given by π-}
    S' ← applyDestroy(S, δ, π-)
    {Apply randomly selected repair operator. Probabilities of repair operators are given by π+}
    S' ← applyRepair(S', π+)
    {Apply local search. Penalty factors are updated based on a dynamic mechanism}
    S' ← applyLocalSearch(S')
    if acceptSA(S', S) then
        S ← S'
    end if
    σ||, σ-, σ+ ← updateScores(S')
    π||, π-, π+ ← updateSelectionProbabilities(σ||, σ-, σ+)
end while

```

Fig. 2. Overview of the ALNS algorithm.

of $n(i, j) + 1$ intermediate vertices ($v_0 = i, v_1, \dots, v_{n(i,j)-1}, v_{n(i,j)} = j$) located on arc (i, j) , where each of the vertices marks a change in gradient or acceleration. Let $A(p(i, j))$ denote the arcs of path p . Now, we can use the energy consumption function for ECVs in Eq. (2) to calculate the energy consumption on each arc $a \in A(p(i, j))$ of the path p :

$$b_a(u_j) = P_a(u_j) \cdot t_a \cdot \begin{cases} \phi^d \cdot \varphi^d & \text{if } P_a(u_j) \geq 0 \text{ kilowatts} \\ \phi^r \cdot \varphi^r & \text{if } P_a(u_j) < 0 \text{ kilowatts} \end{cases}$$

Now, given the energy consumptions on each arc of path $p(i, j)$, we can recursively define the energy consumption on arc (i, j) taking into account that (i) recuperation is restricted by the total battery capacity, and (ii) that we are not allowed to have a negative remaining battery level at any point traveling p . To do so, we let y_{v_k} denote the battery level at each intermediate vertex v_k and define:

$$y_{v_0}(u_j) = y_i$$

$$y_{v_k}(u_j) = \begin{cases} \min(y_{v_{k-1}}(u_j) - b_{v_{k-1}v_k}(u_j), B) & \text{if } y_{v_{k-1}}(u_j) - b_{v_{k-1}v_k}(u_j) \geq 0 \\ -\infty & \text{else} \end{cases} \quad k = 1, \dots, n(i, j)$$

$$b_{ij}(u_j) = y_{v_0}(u_j) - y_{v_{n(i,j)}}(u_j)$$

The original Eq. (3) for calculating the fuel consumption of ICCVs can be adapted in analogous fashion. In this way, the energy consumption on an arc (i, j) is defined by means of the energy consumptions on the path $p(i, j)$, which can be used to represent gradients and accelerations in an arbitrarily fine-grained fashion.

Our model uses additional simplifications. We neglect the influence of outside temperatures on battery capacity. A related issue is the need to power the car heating, which requires additional energy in contrast to ICCVs, for which the waste heat is sufficient to ensure a comfortable temperature in the driver cabin. Further, we neither consider the dependency between charge level and battery efficiency (Hoke, Brissette, Maksimovic, Pratt, & Smith, 2011), nor the non-linear relationship between charge level and recharging duration (Marra et al., 2012), which are present in real-world recharging processes.

5. Adaptive large neighborhood search for solving E-VRPTWMF

This section details our ALNS for addressing E-VRPTWMF. LNS, originally introduced by Shaw (1998), iteratively destroys and repairs potentially larger parts of an initial solution in order to gradually improve the solution. Ropke and Pisinger (2006a) proposed Adaptive LNS, an approach that uses several destroy and repair methods and selects the operators in each iteration based on a probability that depends on the previous success of the respective method.

Our ALNS is inspired by the work of Ropke and Pisinger (2006a) but differs regarding several aspects. First, most ALNS implementations operate on feasible solutions (see, e.g., Demir et al., 2012; Ropke & Pisinger, 2006a, 2006b) and do not reinsert removed customers if their insertion leads to a constraint violation. Contrary to this, we always insert all removed customers in each ALNS step and handle the resulting infeasible solutions by means of a penalty mechanism (cp. Dayarian, Crainic, Gendreau, & Rei, 2013; Hemmelmayr, Cordeau, & Crainic, 2012). Second, we select the number of customers for removal from dynamically changing intervals. Third, our ALNS features a local search component, which is used for intensification, while the main purpose of the ALNS component is the diversification of search. Finally, we introduce a new mechanism for the acceptance decision of the ALNS, where solutions with objective function values calculated based on different penalty factors are compared.

Fig. 2 shows a pseudocode overview of our solution method. First, a preprocessing step is applied to determine arcs that cannot be part of a feasible solution (Section 5.1). Next, we generate an initial solution S with a given number of ECVs and ICCVs (Section 5.3). The initial solution and solutions in the improvement phase may be infeasible and are handled through a penalty mechanism (Section 5.2). ALNS and LS are iteratively applied to improve the solution (Section 5.4).

5.1. Preprocessing

In the preprocessing step, we remove all arcs that cannot be part of a feasible solution, i.e., their inclusion leads to a constraint violation. An arc (v, w) is infeasible if one of the following conditions holds:

1. $v, w \in V \wedge q_v + q_w > Q$
2. $v \in V'_0, w \in V'_{N+1} \wedge e_v + s_v + t_{vw} > l_w$
3. $v \in V'_0, w \in V' \wedge e_v + s_v + t_{vw} + s_w + t_{wN+1} > l_0$
4. $v \in V'_0, w \in V' \wedge \alpha_{vw} \geq 0 \wedge b_{vw}(q_w) > B$
5. $v \in V'_0, w \in V' \wedge \forall j \in F'_0, i \in F'_{N+1} : \alpha_{jv}, \alpha_{vw}, \alpha_{wi} \geq 0 \wedge b_{jv}(q_v + q_w) + b_{vw}(q_w) + b_{wi}(0) > B$

Rules 1–3 eliminate arcs because of capacity and time window violations (cp. Schneider, Stenger, & Goeke, 2014). Rules 4 and 5 are specific to E-VRPTWMF and only hold for ECV routes. Rule 4 determines those arcs that cannot be traveled without violating battery capacities, even if only the demand q_w has to be transported on the arc. Note that a gradient $\alpha_{vw} \geq 0$ is necessary as otherwise recuperation might render the considered arc feasible. Rule 5 additionally considers visits to the best possible recharging station before and after the arc. Note that Rule 4 is not contained in Rule 5 as we have to make the additional assumption that all involved gradients α_{jv}, α_{wi} are greater or equal to zero.

As Rules 4 and 5 are only valid for ECV routes, we keep two separate reduced arc sets: The set A^- contains all arcs that are feasible for ICCVs. The set $A_E^- \subseteq A^-$ excludes arcs that are not feasible in ECV routes.

5.2. Generalized cost function and penalty calculation

During the search, we allow infeasible solutions in order to be more flexible in traversing the solution space. The objective function of a solution S is computed by a generalized cost function $f_{\text{gen}}(S)$ (see, e.g., Gendreau, Hertz, & Laporte, 1994):

$$f_{\text{gen}}(S) = f(S) + \gamma_{\text{cap}} \cdot L_{\text{cap}}(S) + \gamma_{\text{tw}} \cdot L_{\text{tw}}(S) + \gamma_{\text{batt}} \cdot L_{\text{batt}}(S).$$

$f(S)$ denotes the considered objective function: total traveled distance $f_d(S)$ (Eq. 4), routing costs without battery depreciation $f_c(S)$ (Eq. 5), or routing costs including battery depreciation $f_{\text{cb}}(S)$ (Eq. 6). Violations of capacity $L_{\text{cap}}(S)$, time windows $L_{\text{tw}}(S)$ and battery capacity $L_{\text{batt}}(S)$ are scaled by penalty factors γ_{cap} , γ_{tw} and γ_{batt} .

The penalty factors are dynamically adjusted as follows: If a constraint has been violated for β_{pen} iterations, the respective penalty factor is multiplied by a factor ϱ to guide the search towards feasibility. In analogous manner, the respective penalty factor is divided by ϱ if the constraint is met for β_{pen} iterations. The penalty factors are initially set to $\gamma_{\text{cap}}^0 = \gamma_{\text{tw}}^0 = \gamma_{\text{batt}}^0 = 100$ and are restricted to the intervals $[\gamma_{\text{cap}}^{\min}, \gamma_{\text{cap}}^{\max}]$, $[\gamma_{\text{tw}}^{\min}, \gamma_{\text{tw}}^{\max}]$ and $[\gamma_{\text{batt}}^{\min}, \gamma_{\text{batt}}^{\max}]$ respectively. In order to scale the penalty factors to the problem setting and the order of magnitude of the objective function value, we set $\gamma_{\text{cap}}^{\max}$, $\gamma_{\text{tw}}^{\max}$ and $\gamma_{\text{batt}}^{\max}$ to the objective function value of the initial solution. The lower limits $\gamma_{\text{cap}}^{\min}$, $\gamma_{\text{tw}}^{\min}$ and $\gamma_{\text{batt}}^{\min}$ are set to 0.5.

The efficient calculation of changes in constraint violations caused by an LS move is crucial for the quality of our algorithm. Vehicle capacity violations are not affected by the energy consumption model considered in this paper and therefore changes in capacity violations can be calculated in constant time as described in Kindervater and Savelsbergh (1997). However, the considered energy consumption model influences the calculation of changes in battery capacity and time window violations for ECV routes. ICCV routes are also affected if the objective functions f_{cb} or f_c are used.

Let a sequence of customers $\langle v_0, v_1, \dots, v_n, v_{n+1} \rangle$, with v_0 and v_{n+1} representing the depot, define a route r . A solution S is defined as the union of the set of ECV routes $S_E = \{r_k, k = 1, \dots, m_E\}$ and the set of ICCV routes $S_{\text{IC}} = \{r_k, k = m_E + 1, \dots, m_E + m_{\text{IC}}\}$, i.e., $S = S_E \cup S_{\text{IC}}$. Let $\text{Vert}(r)$ denote the set of vertices that are part of route r .

We are interested in the constraint violations of a route that is constructed by concatenating two partial routes $\langle v_0, \dots, u \rangle$ and $\langle w, \dots, v_{n+1} \rangle$ or by inserting a vertex v between the two partial routes. To calculate battery capacity violations of ECV routes, we define the following two variables for each vertex of a route r : $\Upsilon_{v_i}^{\rightarrow}$ is the battery charge that is needed to travel either from the previous visit to a recharging station or from the depot (in case no recharging visit is part of the route) to vertex v_i . $\Upsilon_{v_i}^{\leftarrow}$ is the battery charge that is needed to travel from v_i to either the next recharging station or the depot. The variables are calculated as follows:

$$\Upsilon_{v_i}^{\rightarrow} = \begin{cases} \max(b_{v_{i-1}v_i}(u_{v_i}), 0) & \text{if } v_{i-1} \in F'_0 \\ \max(\Upsilon_{v_{i-1}}^{\rightarrow} + b_{v_{i-1}v_i}(u_{v_i}), 0) & \text{if } v_{i-1} \notin F'_0 \end{cases} \quad i = 1, \dots, n+1$$

$$\Upsilon_{v_i}^{\leftarrow} = \begin{cases} \max(b_{v_i v_{i+1}}(u_{v_{i+1}}), 0) & \text{if } v_{i+1} \in F'_{n+1} \\ \max(\Upsilon_{v_{i+1}}^{\leftarrow} + b_{v_i v_{i+1}}(u_{v_{i+1}}), 0) & \text{if } v_{i+1} \notin F'_{n+1} \end{cases} \quad i = 0, \dots, n.$$

The battery capacity violation of a solution S can be calculated by summing up the individual violations at every visit to a recharging station and on return to the depot:

$$L_{\text{batt}}(S) = \sum_{r_k \in S_E} \left(\sum_{v_i \in \text{Vert}(r_k) \cap F'_{N+1}} \max(\Upsilon_{v_i}^{\rightarrow} - B, 0) \right).$$

If energy consumption is independent of cargo load, changes in battery capacity violation can be calculated in $\mathcal{O}(1)$ for conventional neighborhood operators (see Schneider, Stenger, & Goeke, 2014). If load is considered, concatenation of partial routes or inserting a customer vertex between partial routes alter the energy consumption of all arcs preceding the point of route concatenation or vertex insertion. Thus, individual battery capacity violations at each recharging visit before and the next visit directly after the point of change are modified and therefore necessitate a recalculation. To lower the computational burden, we substitute a surrogate violation $\tilde{L}_{\text{batt}}(S)$ for the real battery capacity violation $L_{\text{batt}}(S)$. The surrogate assumes that violations at the preceding recharging visits remain unmodified. In this way, we are able to compute surrogates of the battery capacity violations in $\mathcal{O}(1)$. Note that the insertion of a recharging station can be handled in constant time without using surrogates.

Time window violations are calculated based on the principle of time travel described in Nagata, Bräysy, and Dullaert (2010); Schneider, Sand, and Stenger (2013), which allows to compute the change in time window violation for conventional inter-route moves in $\mathcal{O}(1)$. In short, the violation of a time window is counted once at the vertex v where the violation occurs and for the calculation of consecutive violations the vehicle is assumed to start service at the latest feasible moment l_v . Schneider, Stenger, and Goeke (2014) adapt this principle to the E-VRPTW. Here, if the partial route $\langle w, \dots, v_{n+1} \rangle$ contains a recharging station z , i.e., $\langle w, \dots, z, z+1, \dots, v_{n+1} \rangle$, variables have to be recalculated for $\langle v, \dots, z+1 \rangle$ in case of vertex insertion and for $\langle w, \dots, z+1 \rangle$ for route concatenation.

In case of the E-VRPTWME, for both vertex insertion and route concatenation, a recalculation from the beginning of the route to the first recharging station contained in the second partial route, i.e., of the partial route $\langle v_0, \dots, z+1 \rangle$, becomes necessary. This is because the amount of required energy (affected by the change in mass along the route as explained above) is needed to calculate recharging times. We again use a surrogate $\tilde{L}_{\text{tw}}(S)$ to reduce computational effort and assume time window violations of vertices preceding the point of change to remain unmodified.

The generalized cost function using surrogates for both battery capacity violation and time window violation is denoted as \tilde{f}_{gen} .

5.3. Generation of initial solution

We generate an initial solution with $m = m_E + m_{\text{IC}}$. First, we build m ECV routes, of which m_{IC} routes are later converted to ICCV routes. Each route is initialized with a seed customer. The seed customers are determined by first sorting the customers in ascending order of their latest start time of service l_v and then selecting the first m entries. The remaining customers in the list are successively inserted into the existing routes at the cost-minimal position according to the generalized cost function $f_{\text{gen}}(S)$. As soon as a route becomes battery capacity infeasible, recharging visits are added at cost-minimal positions (according to $f_{\text{gen}}(S)$) until the route again becomes battery feasible.

After all customers are inserted into routes, we convert long routes with strong constraint violations to ICCV routes by removing recharging visits. More precisely, we sort the routes in descending order of their contribution to the generalized cost function and select the first m_{IC} entries for conversion.

5.4. Solution improvement

The steps of the solution improvement phase are shown in Fig. 2. First, we select the number of customers to be removed in the ALNS step. This number is commonly defined as percentage of the overall number of customers N and is usually drawn from a given interval $[\Omega_{\min}, \Omega_{\max}]$. In preliminary studies, we found that the size of this interval influences the quality of the solution and that a good choice for the interval is instance-dependent. Therefore, we use an adaptive mechanism to select promising intervals (in analogous fashion to the selection of destroy and repair operators). To this end, we split the original interval in ω subintervals without overlap. ALNS then chooses a subinterval according to the probabilities π_i^{\parallel} , $i = 1, \dots, \omega$. From the subinterval, we draw a random number δ of customers and/or stations. Afterwards, we consecutively apply a destroy and repair operator (Section 5.4.1).

The resulting solution is improved by an LS step (Section 5.4.2), which determines a locally optimal solution S' using dynamic penalties as described in Section 5.2. Solution acceptance is based on a Simulated Annealing (SA) criterion (Section 5.4.3). Finally, selection probabilities of the destroy and repair operators, and of the intervals for the numbers of customers to remove are adjusted (Section 5.4.4). To find a feasible solution, the search is run for at most β_{feas} iterations. After a feasible solution is found, the search continues for another β_{obj} to improve the solution.

5.4.1. Destroy and repair operators

Our ALNS uses the following destroy operators:

Random removal removes arbitrary vertices. We implement two versions, the first only removes customers, the second removes customers and recharging visits.

Worst removal was introduced by Ropke and Pisinger (2006b) and aims at removing vertices that are unfavorable at their current position in a route. We sort all vertices in descending order of their contribution to the cost of the current solution, which is determined as the change in \tilde{f}_{gen} that is caused by removing the respective vertex. From this list the vertex at position $\lfloor D \cdot b^{\chi_{\text{worst}}} \rfloor$ is chosen, where D is the size of the list, b is a uniform random number $\in [0, 1]$ and χ_{worst} a parameter to weight the impact that the change in objective value has on the selection. Again a version considering customers and a version considering customers and recharging visits is implemented.

Shaw removal was introduced in Shaw (1997). The idea is to remove customers that are similar to each other with respect to several criteria and are thus likely to be interchangeable. We define the relatedness between two customers i and j by geographical distance d_{ij} , difference in demand $|q_i - q_j|$ and difference of the earliest start time of service $|e_i - e_j|$. Each criterion is weighted with a parameter χ and normalized. The relatedness can be calculated as:

$$R(i, j) = \chi_d \frac{d_{ij}}{\max_{i, j \in V} (d_{ij})} + \chi_q \frac{|q_i - q_j|}{\max_{i \in V} (q_i) - \min_{i \in V} (q_i)} + \chi_e \frac{|e_i - e_j|}{\max_{i \in V} (e_i) - \min_{i \in V} (e_i)}.$$

The first customer to be removed is selected randomly. Then, in each iteration, one customer is drawn randomly from the already removed customers and all non-removed customers are sorted in ascending order of their relatedness value to the selected customer (a small value of $R(i, j)$ corresponds to a high relatedness). From this list, the customer at position $\lfloor D \cdot b^{\chi_{\text{shaw}}} \rfloor$ is chosen, where D is

the size of the list, b is a uniform random number $\in [0, 1]$ and χ_{shaw} a parameter to weight the impact that relatedness has on the selection. Shaw removal is only defined for customer vertices.

Cluster removal identifies and removes clusters of geographically close vertices and was proposed in Ropke and Pisinger (2006b). First, a route is selected at random and vertices contained in the route are separated into two clusters. To this end, Kruskal's algorithm for the minimal spanning tree problem is stopped when two subtrees remain. We randomly select one of the trees as cluster for removal. For the next clustering step, we determine the route closest to a randomly selected customer of the removed cluster. This process is repeated until the required number of vertices is removed. If the overall number of customers in a cluster exceeds the number of vertices that are still to be removed, we randomly select customers from the cluster until the given number is reached. Only one version of cluster removal exists, which does not distinguish between customers and recharging visits.

Station vicinity removal is a special type of the cluster removal operator, which aims at reordering customer visits in the vicinity of recharging stations. We observed that routes in these regions tend to be complex and intertwined because recharging and customer visits have to be scheduled in a manner respecting time window requirements and battery capacity level. To define the vicinity of a station, we randomly choose a radius r in the interval $[\chi_{\text{radius}}^{\min} \cdot \max_{i, j \in V} (d_{ij}), \chi_{\text{radius}}^{\max} \cdot \max_{i, j \in V} (d_{ij})]$, i.e., the radius r is determined instance-dependent as percentage $\chi_{\text{radius}}^{\min} (\chi_{\text{radius}}^{\max})$ of the maximal distance between two customers. Next, we randomly select a recharging station R and remove all customers i with a distance $d_{iR} < r$. This step is repeated until q customers are removed. We implement a second version that also removes all visits to the considered recharging stations.

The following repair operators are used in our ALNS:

Greedy insertion iteratively performs the best possible insertion. For each still unassigned vertex the increase of the surrogate cost function \tilde{f}_{gen} for the insertion at every route position is evaluated and the vertex and position with minimal cost increase are chosen.

Regret insertion was described by Ropke and Pisinger (2006b) in the context of ALNS. The idea is to anticipate the future effect of an insertion operation. The k -regret value for each vertex is calculated as the difference between the insertion cost in the best route and in the k -best route, i.e., k describes the extent to which the future is anticipated. The cost of insertion is evaluated with \tilde{f}_{gen} under the assumption that the insertion is performed at the best possible position within the route. We implement the regret-2 and regret-3 heuristic.

Greedy and Regret insertion are both realized in four variants: the first version only inserts customer vertices, the second version inserts customer vertices and recharging visits, and versions three and four additionally use a diversification penalty to evaluate the costs of inserting a vertex. The diversification penalty is based on the continuous diversification principle introduced in (Cordeau, Laporte, & Mercier, 2001) and later adapted to the E-VRPTW by Schneider, Stenger, and Goeke (2014).

GRASP insertion is based on the GRASP metaheuristic introduced in Feo and Resende (1989). Analogous to greedy insertion, the cost increase for the insertion of all remaining vertices

is calculated and the vertices are stored in a list of size D in ascending order of cost increase. Instead of selecting the best vertex for insertion, a random vertex of the first $\lfloor \chi_{\text{grasp}} \cdot D \rfloor$ best vertices is selected, where χ_{grasp} denotes the degree of randomness. GRASP only inserts customer vertices.

5.4.2. Local search

The LS follows a best-improvement strategy using a composite neighborhood of Relocate (Savelsbergh, 1992), Exchange (Savelsbergh, 1992), 2-opt* (Potvin & Rousseau, 1995) and an operator for inserting and removing visits to recharging stations, called stationInRe (Schneider, Stenger, & Goeke, 2014). Relocate and Exchange are applied for intra- and inter-route moves, 2-opt* is only defined for inter-route moves. Moreover, Relocate is defined to handle recharging visits and 2-opt* allows to modify routes including recharging visits, while Exchange is explicitly not applied to recharging visits. 2-opt* is not allowed between ECV and ICCV routes. Relocate moves inserting recharging visits into ICCV routes are forbidden and also stationInRe moves on ICCV routes.

Moves are realized by means of the generator arc principle introduced in Toth and Vigo (2003). Moves leading to the inclusion of an infeasible arc, i.e., moves that insert an arc from the set $A \setminus A^-$ into an ICCV route or moves inserting an arc from the set $A \setminus A_-^+$ into an ECV route, are not evaluated. As described in Section 5.2, we use a surrogate cost function whose evaluation is less demanding (see, e.g., Crainic, Gendreau, Soriano, & Toulouse, 1993; Gendreau & Potvin, 2010).

In each iteration of the LS, we determine a list \mathcal{M} of the best ϵ solutions that can be generated in the composite neighborhood. The costs used to determine whether a solution is included in the list are either surrogate or exact cost values that were determined in previous iterations (Due to caching, exact values are available for moves that have been evaluated in previous iterations and that only affect routes which have not been changed since the exact evaluation). Then, for each solution with surrogate cost included in the list, we evaluate the exact cost and carry out the best of the ϵ moves based on exact costs.

The LS stops as soon as a local optimum (based on the current penalty factors) is reached.

5.4.3. Acceptance criterion

In order to overcome local optima, ALNS uses an SA-based acceptance criterion (Kirkpatrick, Gelatt, & Vecchi, 1983). Besides always accepting an improving solution, SA accepts a new deteriorating solution with a certain probability that depends (i) on the difference between cost values of the new and the current solution and (ii) a temperature which follows a predefined cooling schedule.

As described above, we allow infeasible solutions and handle them with a dynamic penalty mechanism. Consequently, the cost value of a solution is strongly dependent on the current values of the penalty factors. The dynamic nature of the penalty factors raises the question which penalty factors to use in order to compare the cost value of the current and the new solution. Evaluating both solutions with different penalty factors, e.g., the ones that were valid at the time the solutions were created, can be misleading because differences in cost values may be solely caused by the difference in penalty factors.

Let $(\gamma_{\text{cap}}^S, \gamma_{\text{tw}}^S, \gamma_{\text{batt}}^S)$ be the vector of penalty factors used for evaluating $f_{\text{gen}}(S)$ and $(\gamma_{\text{cap}}^{S'}, \gamma_{\text{tw}}^{S'}, \gamma_{\text{batt}}^{S'})$ the penalty factors used for evaluating $f_{\text{gen}}(S')$ at the time of solution creation. Then, in order to make solutions comparable, the cost values $f_{\text{gen}}^{\text{eval}}(S)$ and $f_{\text{gen}}^{\text{eval}}(S')$ to compute the acceptance probability are determined with penalty factors

$$(\gamma_{\text{cap}}^{\text{eval}}, \gamma_{\text{tw}}^{\text{eval}}, \gamma_{\text{batt}}^{\text{eval}}) = \left(\frac{1}{2} \cdot (\gamma_{\text{cap}}^S + \gamma_{\text{cap}}^{S'}), \frac{1}{2} \cdot (\gamma_{\text{tw}}^S + \gamma_{\text{tw}}^{S'}), \frac{1}{2} \cdot (\gamma_{\text{batt}}^S + \gamma_{\text{batt}}^{S'}) \right).$$

If solutions stay infeasible for longer periods, penalty factors and thus the absolute difference between cost values of solutions increase. To avoid the undesired effect that these (potentially better) solutions have a low chance to be accepted because of high penalty factors, we use the relative difference between cost values to calculate the acceptance probabilities:

$$\Delta f_{\text{rel}}(S', S) = \frac{f_{\text{gen}}^{\text{eval}}(S') - f_{\text{gen}}^{\text{eval}}(S)}{f_{\text{gen}}^{\text{eval}}(S)}.$$

Thus, we always accept improving solutions, while, given the current temperature T , deteriorating solutions are accepted with probability:

$$p(S, S', T) = e^{-\frac{\Delta f_{\text{rel}}(S', S)}{T}}.$$

The starting temperature is chosen in a fashion that a solution that deteriorates the initial solution by ι percent is accepted with a probability of 50 percent. The temperature is decreased by a constant factor such that the temperature is below $T_{\text{threshold}} = 0.0001$ in the last 20 percent of iterations.

5.4.4. Adaptive mechanism

Adaptive weight adjustment assesses the importance of each destroy and repair operator by modifying the probability π with which the respective operator is chosen based on the previous performance of the operator. In more detail, we periodically adjust the probabilities of the destroy operators (π_i^-), of the repair operators (π_i^+), and of the intervals from which the number of customers to remove is drawn (π_i^{\parallel}).

Performance is measured by assigning scores σ_i to the destroy and repair operators and the intervals. If an operator or interval i is used in the current iteration of ALNS, we increase the respective score by σ_A if the following LS results in a new global best solution, by σ_B if a new improving solution is found, and by σ_C if a new deteriorating solution is found and accepted by the SA-based criterion. We consider a solution to be new if we did not obtain the cost value $f_{\text{gen}}(S')$ before (evaluated with a standard set of penalty factors $\gamma_{\text{cap}}, \gamma_{\text{tw}}, \gamma_{\text{batt}} = 1$).

After β_π iterations, we update the weights λ_i^{\parallel} , λ_i^- and λ_i^+ to:

$$\lambda_i := w \frac{\sigma_i}{n_i} + (1 - w) \lambda_i,$$

where w denotes the weight of the current phase of evaluation, $(1 - w)$ the weight put on the previous success of the operator or interval, and n_i the number of times the operator or interval was selected during the last β_π iterations. Then, the scores of each operator or interval are reset to 0. Subsequently, we determine the corresponding probabilities π_i^{\parallel} , π_i^- and π_i^+ to:

$$\pi_i = \frac{\lambda_i}{\sum_i \lambda_i}.$$

6. Numerical studies

We conduct numerical experiments on two different types of instances: First, we use newly generated E-VRPTWMF instances to investigate (i) the influence of using the surrogate objective function on the solution process, (ii) the effect of considering the actual distribution of load in comparison to assuming constant load over all traveled arcs, and (iii) the influence of different objective functions on the structure and costs of generated solutions and the share that ECVs contribute to this cost. Second, to assess the performance of ALNS in terms of solution quality and run-time, we conduct tests on available benchmark instances of the related problems VRPTW and E-VRPTW.

The section is structured as follows. The parameter tuning is presented in Section 6.1, the generation of the E-VRPTWMF benchmark instance in Section 6.2, the tests performed on these instances in Section 6.3, and the tests on the benchmark instances of related problems in Section 6.4.

Table 2

Results of different parameter settings on a subset of the E-VRPTWMF instances. The best setting for each of the parameters is marked in bold and used as final setting. Results for the tested settings for each parameter are given as deviation (in percent) of the best solution achieved in 10 runs with the tested setting to the result obtained with the best setting for the respective parameter (Δ_{fd}).

ALNS			
β_π	25	50	75
Δ_{fd}	0.03	0.02	0.0
$(\sigma_A, \sigma_B, \sigma_C)$	(20, 10, 10)	(10, 20, 10)	(5, 5, 10)
Δ_{fd}	0.0	0.0	0.10
w	0.2	0.4	0.6
Δ_{fd}	0.28	0.23	0.0
Ω_{\max}	0.3	0.4	0.5
Δ_{fd}	0.04	0.11	0.0
$(\chi_{\text{radius}}^{\min}, \chi_{\text{radius}}^{\max})$	(0.01, 0.15)	(0.05, 0.15)	(0.5, 0.2)
Δ_{fd}	0.13	0.01	0.00
ι	0.1	0.3	0.5
Δ_{fd}	0.02	0.0	0.08
LS			
ϱ	1.2	1.5	2.0
Δ_{fd}	0.10	0.14	0.0
ϵ	25	50	75
Δ_{fd}	0.0	0.10	0.09

Table 3

Final parameter setting of ALNS used in the numerical studies.

ALNS		ALNS		LS	
β_{feas}	2000	$(\Omega_{\min}, \Omega_{\max})$	(0.1, 0.5)	ϵ	25
$\beta_{\text{obj}} (N \leq 100)$	750	ω	5	$\gamma_{\text{cap}}^0, \gamma_{\text{tw}}^0, \gamma_{\text{batt}}^0$	100
$\beta_{\text{obj}} (N > 100)$	$750/(0.01 \cdot N)^{1.5}$	$(\chi_{\text{radius}}^{\min}, \chi_{\text{radius}}^{\max})$	(0.05, 0.15)	$\gamma_{\text{cap}}^{\min}, \gamma_{\text{tw}}^{\min}, \gamma_{\text{batt}}^{\min}$	0.5
β_π	75	χ_{grasp}	0.2	$\gamma_{\text{cap}}^{\max}, \gamma_{\text{tw}}^{\max}, \gamma_{\text{batt}}^{\max}$	$f_{\text{gen}}(S_{\text{initial}})$
$(\sigma_A, \sigma_B, \sigma_C)$	(10, 20, 10)	χ_{worst}	6	ϱ	2.0
w	0.6	χ_{shaw}	6	β_{pen}	2
ι	0.3	(χ_d, χ_q, χ_e)	(6, 4, 5)		

6.1. Experimental environment and parameter setting

All experiments are conducted on a desktop computer with an Intel Core i7 processor at 2.8 gigahertz, 8 gigabytes of RAM, running Windows 7 Enterprise. Our ALNS is implemented as single-thread code in Java. The parameter setting of the ALNS is determined using a subset of 20 randomly selected E-VRPTWMF instances with 50 and 100 customers (details on the generation of E-VRPTWMF instances are given in Section 6.2). Parameter tests are carried out using the objective of minimizing traveled distance.

During the development of our algorithm, we identified (i) a set of parameters that have a stronger influence on the solution quality of the algorithm, and (ii) for each of these parameters a well-performing base setting. In the parameter tuning, we investigate the influence of modifying this base setting in two directions, i.e., three settings are tested per parameter. More precisely, we consecutively assess each of the parameters and always keep the best setting found for this parameter for the tuning of the remaining parameters. The order in which the parameters are tuned is determined randomly. Table 2 shows, for each parameter, the three tested settings and the deviation Δ_{fd} of the best solution found in 10 runs with the respective setting to the result obtained with the best setting for this parameter. The best setting for each parameter is given in bold.

In the described fashion, we test the following parameters: the number of iterations after which the weights of ALNS are adjusted (β_π), the scores used in the adaptive weight adjustment ($\sigma_A, \sigma_B, \sigma_C$), the reaction factor (w) of the weight adjustment, the maximal percentage of removed customers (Ω_{\max}), the minimal and maximal radius for the station vicinity removal operator ($\chi_{\text{radius}}^{\min}, \chi_{\text{radius}}^{\max}$), the percentage of deterioration that is accepted at the beginning of SA (ι), the penalty update factor (ϱ), and the number of solutions in each

iteration that are evaluated with the original objective function instead of the surrogate (ϵ). We observe that none of the tested settings leads to a significant deterioration of solution quality (the deviation is always below 0.28 percent). Consequently, our ALNS seems relatively insensitive to parameter variations.

In order to achieve a good tradeoff between run-time and solution quality, we set the maximal number of ALNS iterations to find a feasible solution to $\beta_{\text{feas}} = 2000$ and the number of iterations for further improvement to $\beta_{\text{obj}} = 750$ for all instances with less than 100 customers. For larger instances, we use $\beta_{\text{obj}} = 750/(0.01 \cdot |N|)^{1.5}$ in order to keep run-times acceptable. Table 3 summarizes the final parameter setting, which is used for all of the following experiments.

6.2. Generation of E-VRPTWMF instances

This section describes the generation of E-VRPTWMF test instances. The instances are based on the PRP instances proposed in Demir et al. (2012). The customer locations in these instances represent real cities in the UK, customer demands and time windows are generated in random fashion. The benchmark consists of nine instance sets grouped according to the size of the instances contained in each set (10–200 customers). Each set contains 20 instances with the respective size. Contrary to the PRP, speed is not a decision variable in E-VRPTWMF and we set a fixed speed that corresponds to the upper speed limit given in the PRP instances, i.e. we set $v_{ij} = 90$ kilometers per hour, $\forall i, j \in V_{0,N+1}^*$. In this way, we also ensure that all instances can be feasibly solved (with respect to time windows) when using a fleet of ICCVs.

The number of recharging stations is adapted to the number of customers N of the instances. More precisely, we locate $\lfloor 0.1 \cdot |N| \rfloor$ recharging stations at randomly drawn customer locations and one

Table 4
Data of E-VRPTWMF instances.

Notation	Value	Description	Source
g	9.81 meters per square second	Gravitational constant	Demir et al. (2012)
ρ	1.2041 kilograms per cubic meter	Air density	Demir et al. (2012)
A	3.912 square meters	Frontal surface	Demir et al. (2012)
m_c	6350 kilograms	Curb mass	Demir et al. (2012)
m_u	1 kilogram	Mass per unit	
Q	3650	Capacity	Demir et al. (2012)
c_r	0.01	Coefficient rolling resistance	Demir et al. (2012)
c_d	0.7	Coefficient aerodynamic drag	Demir et al. (2012)
k	0.2 kilojoules per revolution per liter	Engine friction factor	Demir et al. (2012)
ξ	1	Fuel-to-air mass ratio	Demir et al. (2012)
η	0.9	Efficiency parameter diesel engines	Demir et al. (2012)
η_{df}	0.4	Efficiency parameter drive train	Demir et al. (2012)
κ	44 kilojoules per gram	Heating value diesel fuel	Demir et al. (2012)
ψ	737 liters per gram	Conversion factor	Demir et al. (2012)
N	33 revolutions per second	Engine speed	Demir et al. (2012)
D	5 liters	Engine displacement	Demir et al. (2012)
B	80 kilowatt-hour	Battery capacity	Davis and Figliozzi (2013)
r	120^{-1} hours per kilowatt	Recharging rate	Tesla Motors, Inc. (2014)
ϕ^d	1.184692	Efficiency parameter (motor mode)	See Section 3.1
ϕ^r	0.846055	Efficiency parameter (generator mode)	See Section 3.1
φ^d	1.112434	Efficiency parameter discharging	See Section 3.1
φ^r	0.928465	Efficiency parameter recuperation	See Section 3.1
c^D	13.1368 dollars per hour	Driver wages, conversion ratio 1.6421 \$/€	Demir et al. (2012)
c^{IC}	1.07 dollars per liter	Diesel fuel cost	Davis and Figliozzi (2013)
c^E	0.1106 dollars per kilowatt-hour	Electricity cost	Davis and Figliozzi (2013)
c^B	0.1989 dollars per kilometer	Battery replacement cost	See below

additional station at the depot. To determine the number of ECVs and ICCVs of each instance, we proceed as follows: We start with a total vehicle number m that corresponds to the number of ICCVs given in the PRP solutions of Demir et al. (2012), and gradually substitute ICCVs with ECVs until either $m_E = \lfloor m/2 \rfloor$, or our ALNS is no longer able to find a feasible solution.

We use the physical constants, vehicle properties and coefficients provided in Demir et al. (2012). The battery capacity is set to 80 kWh, which is a realistic for ECVs according to Davis and Figliozzi (2013). Efficiency parameters for engine and battery during discharge and recuperation are determined as described in Section 3.1. The recharging power at stations is assumed equivalent to that of the Tesla supercharger network for passenger cars (Tesla Motors, Inc., 2014).

When using the objectives f_c or f_{cb} , we assume the following cost structure: Driver costs are equal for ICCVs and ECVs and correspond to the time spend en route. Hourly wages are given in Demir et al. (2012). The diesel price, electricity price and costs for battery replacement used for calculating driving costs are taken from Davis and Figliozzi (2013). To determine the battery replacement cost on a per kilometer basis, we assume that a battery has to be replaced after 241,350 kilometers (150,000 miles) at a cost of 600 dollars per kilowatt-hour. Table 4 summarizes the data. The complete instances are available for download at http://www.logistikplanung.tu-darmstadt.de/logistikplanung_und_informationssysteme/forschung_16/publikationen_14/index.de.jsp.

6.3. Experiments on E-VRPTWMF instances

We conducted tests with our ALNS for the complete set of instances described above. The detailed results are given in the appendix in Appendix A as comparison values for researchers addressing the same instances in the future. In this section, we describe the numerical tests that investigate the research questions introduced above:

1. impact of using the surrogate objective function (Section 6.3.1)
2. effect of considering load in the calculation of energy consumption (Section 6.3.2)

3. influence of different objective functions and the resulting level of usage of ECVs (Section 6.3.3).

6.3.1. Influence of surrogate cost function

As described in Section 5.4.2, the LS determines the best move in each iteration based on exactly evaluating a list \mathcal{M} of promising candidate moves that were found using a surrogate cost function \tilde{f}_{gen} . We assess the impact of using the surrogate cost function (i) on the course of the search, and (ii) on the solution quality of our ALNS.

First, we conduct 50 iterations of ALNS on the 100-customer instance E-UK_100_03 with the objective of minimizing traveled distance. The size of the list \mathcal{M} is set to $\epsilon = 50$. In each iteration, we calculate the best move based on the surrogate cost function, i.e., we fill \mathcal{M} using the surrogate cost, then determine the exact costs of the entries in \mathcal{M} and select the best entry. The quality of this move is assessed by determining its rank among the possible moves in this iteration if all moves were evaluated exactly (not only those in \mathcal{M}).

In Fig. 3, we plot these ranks. Applications of destroy and repair operators are marked with an “x”. Thus, we identify how often and to what degree we miss the best possible move. In 45 percent of iterations we choose the rank 1 move and in 59 percent of iterations we choose one of the top 3 moves. Only in 3.5 percent of iterations, the selected move is not among the top 10.

Second, we carry out additional experiments to investigate the effect of using the surrogate cost function on the solution quality of ALNS by comparing the solutions obtained with the surrogate and with the exact cost function. To keep computation times reasonable, the tests are run on a reduced subset of the E-VRPTWMF instances, which is obtained by randomly selecting 5 out of 20 instances for each problem size. Due to the very high run-times when using the exact cost function, we further have to restrict the size of the considered instances to maximally 75 customers for this study. Summarizing, we randomly select 5 of the 20 instances for the problem sizes $|N| = 5, 10, 15, 20, 25, 50, 75$ and conduct 10 runs for each instance with the surrogate cost function and with the exact cost function, following the objective of minimizing traveled distance. The size of the list \mathcal{M} is again set to $\epsilon = 50$.

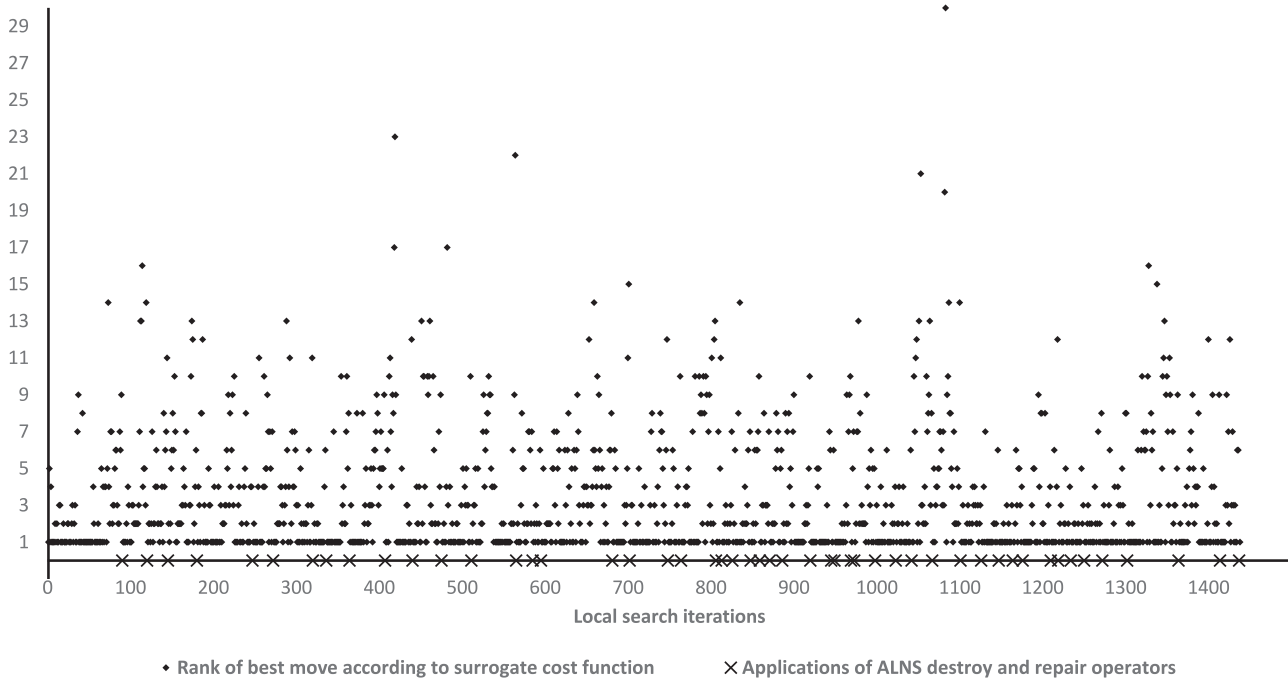


Fig. 3. Effect of using the surrogate cost function \tilde{f}_{gen} to determine the best move in 50 ALNS iterations on the 100-customer instance E-UK_100_03. The figure depicts the rank of the selected best move among all possible moves evaluated with the exact cost function.

Table 5

Comparison of the solution quality using the exact cost function and using the surrogate cost function. Run-time t_{avg} is reported in minutes, the gaps Δ_d^{best} and Δ_d^{avg} in percent.

Inst. group	Exact cost function			Surrogate cost function		
	f_d^{best}	f_d^{avg}	t_{avg}	Δ_d^{best}	Δ_d^{avg}	\mathcal{T}
E-UK10	456.60	456.60	0.13	0.00	0.00	6.45
E-UK15	663.04	663.06	0.73	0.00	0.00	13.94
E-UK20	837.71	837.76	1.97	0.00	0.00	21.41
E-UK25	940.80	941.40	6.07	0.00	−0.03	27.81
E-UK50	1408.05	1415.55	100.08	0.08	0.38	107.94
E-UK75	1932.65	1939.69	419.49	0.15	0.50	194.04

For the exact cost function, Table 5 reports the best solution found in the 10 runs (f_d^{best}), the average solution quality (f_d^{avg}) and the average run-time in minutes (t_{avg}) as averages of groups of instances of the same size. For the surrogate cost function, the same measures are evaluated and the solution quality is reported as gaps Δ_d^{best} and Δ_d^{avg} (in percent) to the solutions obtained with the exact cost function. The run-time using the surrogate cost function is presented using a run-time reduction factor \mathcal{T} , which is obtained by dividing the average run-time using the exact cost function by the average run-time using the surrogate cost function. At the bottom of the table, average results over the investigated instance groups are given.

The results show that using the surrogate function has only minimal negative effect on the solution quality concerning the best as well as the average solution quality. However, the run-time savings using the surrogate are really impressive with a factor \mathcal{T} of 194 for the 75-customer instances.

6.3.2. Effect of considering the distribution of load on solution quality and cost estimates

This study investigates the importance of considering the distribution of load when routing a mixed fleet of ICCVs and ECVs. To this end, we compare the quality of the solutions obtained based on the actual load distribution to the solutions obtained with the following two load estimates. First, the cautious load estimate is to assume that all vehicles are fully loaded during their whole trip ($\bar{u} = Q$). This is a worst-case estimate in some sense because it implies maximal energy consumption. This estimate guarantees that all obtained solutions are

still feasible with respect to battery capacity and time windows if the actual load distribution is considered. The second load estimate assumes the average case that the vehicles travel half-loaded ($\bar{u} = Q/2$). This may result in solutions that are infeasible when taking into account the actual load due to either violations of battery capacity, i.e., ECVs that get stranded, or due to violations of time windows, which are caused by a wrong estimation of the recharging time of ECVs.

We are interested in:

- the quality of solutions S obtained with the estimates $S(\bar{u} = Q)$ and $S(\bar{u} = Q/2)$ when afterwards evaluating the costs considering the actual load (denoted as $f_c^{u_i}(S)$), and
- the error of determined costs when the solutions are obtained and evaluated based on the load estimates (denoted as $f_c^{\bar{u}=Q}(S)$ and $f_c^{\bar{u}=Q/2}(S)$). These cases are interesting as these are the cost estimates that will often be used in practical planning situations.

We conduct experiments addressing the above points on a reduced subset of the E-VRPTWMF instances (5 randomly chosen instances out of 20 for each group), again, in order to keep computation times reasonable. For each instance, 10 ALNS runs are performed following the objective of minimizing costs without battery costs f_c and the reported results are based on the best of the 10 runs. Table 6 provides for each of the instances the number of ICCVs (m_{IC}) and ECVs (m_{E}), the cost of the solution obtained based on the actual load distribution and evaluated using the actual load ($f_c^{u_i}(S(u_i))$), and the percentage gap (to the costs $f_c^{u_i}(S(u_i))$) of the solutions obtained considering load estimates and evaluated using (i) the actual load (columns

Table 6

Effect of considering the distribution of load on the solution quality and cost estimates. For each instance, the number of ICCVs (m_{IC}) and ECVs (m_E) and the results of the best of 10 ALNS runs are reported. $f_c^{u_i}(S(u_i))$ denotes the cost (in dollar) of the solution obtained based on the actual distribution of load and evaluated using the actual load, $\Delta f_c^{u_i}(S(\bar{u} = Q))$ and $\Delta f_c^{u_i}(S(\bar{u} = Q/2))$ the percentage gaps (to the costs $f_c^{u_i}(S(u_i))$) of the solutions obtained based on load estimates and evaluated using the actual load, and $\Delta f_c^{\bar{u}=Q}(S(\bar{u} = Q))$ and $\Delta f_c^{\bar{u}=Q/2}(S(\bar{u} = Q/2))$ the percentage gaps of the solutions obtained based on load estimates and evaluated using load estimates. Instances with an infeasible solution are marked with a dash. Average results over the investigated instances are given.

Inst.	m_{IC}	m_E	$S(u_i)$	$S(\bar{u} = Q)$		$S(\bar{u} = Q/2)$	
			$f_c^{u_i}$	$\Delta f_c^{u_i}$	$\Delta f_c^{\bar{u}=Q}$	$\Delta f_c^{u_i}$	$\Delta f_c^{\bar{u}=Q/2}$
E-UK10_04	1	1	207.14	0.25	4.20	0.25	0.46
E-UK10_09	1	1	188.32	0.06	4.23	0.06	0.39
E-UK10_13	1	1	207.96	0.32	4.52	0.32	0.31
E-UK10_18	1	1	159.86	5.61	9.47	5.61	0.10
E-UK10_20	1	1	165.06	7.60	13.15	7.98	1.35
E-UK15_01	2	0	313.46	0.00	3.38	0.00	−0.33
E-UK15_07	2	1	267.11	0.00	4.36	0.00	0.92
E-UK15_09	2	1	265.72	6.23	9.17	0.09	−4.88
E-UK15_12	2	1	319.79	0.96	5.33	0.96	0.63
E-UK15_15	1	1	231.21	12.64	18.01	0.00	−10.66
E-UK20_05	2	1	316.31	3.00	6.29	2.38	−0.79
E-UK20_06	2	1	366.85	0.36	3.64	0.04	−0.69
E-UK20_09	2	1	337.95	0.00	3.95	0.00	0.36
E-UK20_14	2	2	410.33	6.79	10.92	3.59	−2.74
E-UK20_15	2	1	338.95	0.09	4.07	0.00	0.19
E-UK25_10	2	2	367.30	2.74	6.50	0.26	−2.15
E-UK25_11	2	2	376.18	1.98	5.89	—	−1.54
E-UK25_12	2	2	426.02	2.15	6.15	—	−3.93
E-UK25_16	2	2	348.59	2.61	6.40	0.94	−1.41
E-UK25_19	2	2	441.59	4.07	8.28	—	−2.06
E-UK50_11	4	3	646.02	1.32	5.11	1.03	−0.04
E-UK50_12	4	3	586.88	0.74	4.17	0.24	−0.45
E-UK50_13	4	3	587.82	0.26	3.33	—	−1.21
E-UK50_16	4	3	582.00	0.96	3.95	1.12	−0.07
E-UK50_18	4	4	695.65	1.00	3.89	0.71	−0.63
E-UK75_02	6	5	854.35	2.37	5.20	0.54	−2.00
E-UK75_04	6	5	833.66	0.92	3.79	0.06	−0.76
E-UK75_09	5	5	933.15	1.32	4.69	—	−1.20
E-UK75_12	5	5	859.67	0.57	3.74	0.11	−0.62
E-UK75_17	6	5	914.59	1.09	4.18	—	−0.11
E-UK100_01	7	7	1247.03	0.07	2.91	—	−0.96
E-UK100_03	7	6	1120.41	0.80	3.90	—	−0.52
E-UK100_06	7	7	1237.11	1.59	4.74	0.47	−1.45
E-UK100_07	6	6	1079.63	0.86	3.92	—	−1.40
E-UK100_17	8	7	1308.21	1.43	4.48	—	−0.90
E-UK150_01	10	10	1527.75	1.77	4.35	—	−1.00
E-UK150_02	10	10	1758.03	2.16	5.14	—	−2.23
E-UK150_03	10	9	1535.77	0.88	3.40	0.87	−0.20
E-UK150_13	10	9	1692.97	1.42	4.32	—	−0.92
E-UK150_17	10	10	1728.11	3.09	6.24	—	−2.59
E-UK200_03	14	13	2212.82	0.82	3.68	−0.09	−1.09
E-UK200_05	14	13	2316.20	−0.06	2.69	—	−0.94
E-UK200_08	14	13	2217.84	2.10	4.91	—	0.19
E-UK200_13	13	12	2284.02	0.02	2.98	0.15	−0.10
E-UK200_16	14	13	2183.56	0.10	2.84	—	0.08
Avg.				1.89	5.34	0.99	−0.94

$\Delta f_c^{u_i}(S(\bar{u} = Q))$ and $\Delta f_c^{u_i}(S(\bar{u} = Q/2))$, and (ii) the load estimate (columns $\Delta f_c^{\bar{u}=Q}(S(\bar{u} = Q))$ and $\Delta f_c^{\bar{u}=Q/2}(S(\bar{u} = Q/2))$). Instances for which the solution found with the estimate $\bar{u} = Q/2$ is not feasible if the actual load is considered are marked with a dash. At the bottom of the table, average results over the investigated instances are given.

As mentioned above, using the estimate $\bar{u} = Q$ guarantees feasibility, however, the obtained solutions show an average gap of 1.89 percent if the actual load distribution is used for evaluating costs. Further, cost evaluation based on this estimate leads to an overestimation of costs by more than 5 percent on average. On the other hand, using the estimate $\bar{u} = Q/2$ leads to infeasible solutions for 17 out of 45 instances and an average gap of approximately 1 percent over the

instances for which a feasible solution is found. Costs based on this estimate underestimate real costs by −0.94 percent.

The results show that the quality of solutions obtained using estimates is clearly inferior to the quality of the solutions obtained considering the actual load. Moreover, costs evaluated with estimates strongly differ from real costs. Both deviations are stronger in the case of $\bar{u} = Q$. However, in the context of ECVs, using a load estimate other than this “worst-case estimate” may result in infeasible solutions due to stranded ECVs or time window violations caused by wrong assessment of the required recharging times of ECVs. This is different from ICCV fleet, where load estimates can only lead to wrong cost estimations and wrong routing decisions but not to infeasible solutions. Also

Table 7

Comparison of the different objective functions f_d , f_c and f_{cb} . Results of the best of 10 ALNS runs, averaged over instance groups are reported. S_x , $x \in \{d, c, cb\}$ denotes the solution obtained when minimizing with objective f_x and $f_y(S)$, $y \in \{d, c, cb\}$ denotes the objective value (in dollar, respectively kilometers) of evaluating solution S with objective function f_y . The results for using the cost-minimizing objectives f_c and f_{cb} are given as percentage gaps to the values obtained using the distance-minimizing objective f_d .

Inst. group	S_d			S_c			S_{cb}		
	$f_d(S_d)$	$f_c(S_d)$	$f_{cb}(S_d)$	$\Delta f_d(S_c)$	$\Delta f_c(S_c)$	$\Delta f_{cb}(S_c)$	$\Delta f_d(S_{cb})$	$\Delta f_c(S_{cb})$	$\Delta f_{cb}(S_{cb})$
E-UK10	456.60	198.61	219.65	2.54	−6.50	−3.39	2.14	−2.95	−5.71
E-UK15	663.04	295.67	321.21	1.60	−5.53	−2.37	2.24	−2.82	−3.73
E-UK20	837.71	373.99	418.87	2.08	−5.37	−3.13	1.01	−4.65	−4.76
E-UK25	940.80	404.09	477.41	1.64	−3.02	−1.01	0.77	−2.19	−3.16
E-UK50	1409.00	665.59	766.10	4.16	−6.92	1.22	2.45	−4.52	−4.22
E-UK75	1935.64	937.70	1088.39	2.24	−6.23	−1.78	2.00	−4.19	−5.51
E-UK100	2666.67	1250.15	1475.69	2.84	−4.14	−0.43	1.91	−3.01	−3.46
E-UK150	3487.07	1705.31	2007.53	3.15	−3.34	0.15	1.47	−2.90	−3.04
E-UK200	4671.72	2316.99	2697.74	3.91	−3.20	0.79	3.57	−1.75	−0.94
Avg.	1896.47	905.35	1052.51	2.68	−4.92	−1.11	1.95	−3.22	−3.84

note that it may happen that using a certain load estimate, it is not possible to find a feasible solution although a feasible solution exists for the actual load distribution. For example, this may be the case if time windows are very restrictive and ECVs have to travel with very little load in order to avoid recharging operations.

Overall, the study clearly demonstrates the importance of considering load information in the route planning of ECVs from a feasibility as well as a cost perspective. While the problem of stranded ECVs may be counteracted by allotting high safety margins, an adequate consideration of energy consumption allows to better utilize expensive battery capacity. Finally, we note that our ALNS is capable to solve all instances with the same number of vehicles as determined originally for the PPR and we can substitute approximately half of the ICCVs in the fleet ($m_E/(m_E + m_{IC}) = 47.2$ percent) by ECVs.

6.3.3. Assessment of different objective functions and cost contribution of ECVs

In this study, we compare the solutions obtained with the different objective functions of minimizing traveled distance (f_d), costs without battery depreciation cost (f_c) and costs including battery costs (f_{cb}) in terms of their performance regarding the different objectives. In the second part of the study, we investigate to what extent ECVs are employed when different objective functions are used.

We conduct 10 ALNS runs with each of the objective functions f_d , f_c and f_{cb} on the reduced instance set described above, and we assess the results of the best of the 10 runs for each instance. Table 7 reports the results as averages over the instance groups with different customer numbers 10, 15, 20, 25, 50, 75, 100, 150 and 200. Let S_x , $x \in \{d, c, cb\}$ denote the solution obtained when minimizing with objective f_x and let $f_y(S)$, $y \in \{d, c, cb\}$ denote the objective value of evaluating solution S with objective function f_y . The table reports the average values over the instance group for the solutions S_d obtained with the objective of minimizing traveled distance (columns $f_d(S_d)$, $f_c(S_d)$, and $f_{cb}(S_d)$), and the percentage gaps (to these values) of the objective values of the solutions obtained when minimizing costs (columns $\Delta f_d(S_c)$, $\Delta f_c(S_c)$, and $\Delta f_{cb}(S_c)$) and when minimizing costs including battery costs (columns $\Delta f_d(S_{cb})$, $\Delta f_c(S_{cb})$, and $\Delta f_{cb}(S_{cb})$).

The results show that the ALNS is effective with regard to the different optimization objectives: The best results found with respect to a certain objective function are always found when optimizing according to this objective. The gaps of the solutions found with other objectives are partly significant with nearly 5 percent. In particular, the results show that the traditional objective of minimizing traveled distance does not provide satisfying solutions when costs for fuel (diesel and electricity), labor, and battery are considered. This is in line with the results of Bektaş and Laporte (2011) for pure ICCV fleets, who note similar effects considering the cost components fuel, labor and emission costs. If battery costs are included, the quality of distance-minimizing solutions rises as traveled distance becomes

Table 8

Percentage share that ECVs contribute to the traveled distance of the solutions obtained with the different objective functions.

Inst. group	$d_E(f_d(S_d))$	$d_E(f_d(S_c))$	$d_E(f_d(S_{cb}))$
E-UK10	23.92	29.80	16.18
E-UK15	19.92	25.76	16.57
E-UK20	25.66	29.42	23.94
E-UK25	39.32	42.57	35.71
E-UK50	35.30	53.23	33.82
E-UK75	39.18	48.11	33.00
E-UK100	42.76	49.83	39.41
E-UK150	43.34	50.47	41.07
E-UK200	40.92	49.31	41.11
Avg.	34.48	42.06	31.20

Table 9

Results of ALNS on Solomon benchmark instances in comparison to the currently best VRPTW heuristic from the literature by Nagata, Bräysy, and Dullaert (2010) (NBD). We report the number of vehicles m and the traveled distance f of the best of 10 runs for the ALNS, and of 5 runs for NBD. Average results for each instance group, the cumulative number of vehicles (CNV) and the cumulative traveled distance (CTD) are given.

Inst. group	NBD ($m f$)	ALNS ($m f$)
R1	11.92 1210.34	11.92 1215.60
R2	2.73 951.03	2.73 958.02
C1	10.0 828.38	10.0 828.38
C2	3.0 589.86	3.0 589.86
RC1	11.5 1384.16	11.5 1388.76
RC2	3.25 1119.24	3.25 1123.37
CNV	405	405
CTD	57187	57397

more expensive (there is only a gap of −3.84 percent for $\Delta f_{cb}(S_{cb})$ compared to a gap of −4.92 percent for $\Delta f_c(S_c)$).

In Table 8, we report the percentage share that ECVs contribute to the traveled distance of the solutions obtained with the different objective functions, denoted as $d_E(f(S))$. If ICCVs and ECVs were used to the same extent, ECVs would contribute with $m_E/(m_E + m_{IC}) = 47.2$ percent to the traveled distance. We find that when minimizing traveled distance, ECVs travel a noticeably smaller part of the total distance because long routes will mostly be assigned to ICCVs in order to save detours. The share of traveled distance rises significantly when costs without battery costs are considered because ECV travel is cheaper than ICCV travel if only fuel and electricity consumption is considered. Finally, if costs including battery costs are considered, the contribution of ECVs decreases perceptibly because ECVs become relatively expensive if battery depreciation costs are added to the travel costs.

Table 10

Results of our ALNS on E-VRPTW instances in comparison to those of the ALNS of [Hiemann, Puchinger, and Hartl \(2014\)](#) (HPH), and of the VNS/TS of [Schneider, Stenger, and Goeke \(2014\)](#) (SSG). BKS denotes the previously best known solution. For each algorithm, we report the number of vehicles m and the percentage gap Δ_{best} of the best solution obtained in 10 runs to the BKS. For HPH and ALNS we also report run-times in column t (in minutes). Column ALNS reports the best solutions found during the overall testing of ALNS. The cumulative number of vehicles (CNV), average gaps and average run-times are given at the bottom of the table.

Inst.	BKS		HPH			SSG		ALNS		ALNS	
	m	f	m	Δ_{best}	t	m	Δ_{best}	Δ_{best}	t	f	Δ_f
c101	12	1053.83	12	0.00	9.16	12	0.00	0.00	1.62	1053.83	0.00
c102	11	1056.47	11	0.07	9.76	11	0.07	0.07	2.61	1051.38	−0.48
c103	10	1041.55	10	0.25	9.15	10	0.00	−0.26	3.61	1034.86	−0.64
c104	10	979.51	10	0.52	11.09	10	0.13	−0.76	4.89	961.88	−1.80
c105	11	1075.37	11	0.00	9.31	11	0.00	0.00	1.82	1075.37	0.00
c106	11	1057.65	11	0.00	9.44	11	0.02	0.00	2.44	1057.65	0.00
c107	11	1031.56	11	0.00	9.66	11	0.00	0.00	2.72	1031.56	0.00
c108	10	1100.32	10	0.83	9.21	10	0.00	−0.13	3.20	1095.66	−0.42
c109	10	1036.64	10	1.43	10.17	10	1.47	−0.29	2.84	1033.67	−0.29
c201	4	645.16	4	0.00	18.11	4	0.00	0.00	1.45	645.16	0.00
c202	4	645.16	4	0.21	21.13	4	0.00	0.00	2.49	645.16	0.00
c203	4	644.98	4	0.00	22.91	4	0.00	0.00	3.48	644.98	0.00
c204	4	636.43	4	0.30	19.70	4	0.00	0.00	3.45	636.43	0.00
c205	4	641.13	4	0.00	20.96	4	0.00	0.00	2.03	641.13	0.00
c206	4	638.17	4	0.00	23.26	4	0.00	0.00	2.33	638.17	0.00
c207	4	638.17	4	0.00	22.68	4	0.00	0.00	2.71	638.17	0.00
c208	4	638.17	4	0.00	22.30	4	0.00	0.00	2.61	638.17	0.00
r101	18	1663.04	18	0.00	8.80	18	0.57	0.46	1.58	1665.62	0.16
r102	16	1488.97	16	0.00	9.85	16	3.15	0.26	2.26	1487.41	−0.10
r103	13	1285.96	13	0.00	9.91	13	1.06	−0.31	2.62	1271.35	−1.14
r104	11	1088.43	11	0.86	8.73	11	0.00	0.21	2.43	1088.43	0.00
r105	14	1461.25	15	−1.87	9.31	14	0.84	−0.51	3.38	1442.35	−1.29
r106	13	1344.66	13	1.38	8.91	13	0.00	−1.13	2.64	1324.10	−1.53
r107	12	1154.52	12	0.94	9.16	12	0.00	−0.26	2.56	1150.95	−0.31
r108	11	1050.04	11	1.66	8.78	11	1.51	1.22	2.63	1050.04	0.00
r109	12	1294.05	13	−3.72	9.48	12	0.00	−2.53	3.76	1261.31	−2.53
r110	11	1126.74	11	2.56	9.15	11	1.49	1.30	3.69	1119.50	−0.64
r111	12	1106.19	12	1.29	9.34	12	1.62	1.61	2.71	1106.19	0.00
r112	11	1026.52	11	1.68	8.70	11	0.00	0.02	2.73	1016.63	−0.96
r201	3	1264.82	3	0.37	27.05	3	0.00	0.21	2.05	1264.82	0.00
r202	3	1052.32	3	0.15	27.60	3	0.00	0.05	2.71	1052.32	0.00
r203	3	895.91	3	0.14	27.81	3	1.89	0.35	3.14	895.54	−0.04
r204	2	788.67	2	0.00	17.86	2	0.24	−0.73	4.55	779.49	−1.16
r205	3	988.67	3	1.35	24.85	3	0.00	0.14	2.08	987.36	−0.13
r206	3	922.70	3	0.00	24.98	3	0.27	0.23	2.60	922.19	−0.06
r207	2	848.53	2	1.33	20.83	2	0.49	0.05	3.46	845.26	−0.39
r208	2	736.60	2	0.49	18.25	2	0.00	0.44	3.52	736.12	−0.07
r209	3	872.36	3	2.10	25.56	3	0.00	0.16	2.66	867.05	−0.61
r210	3	847.06	3	1.94	25.56	3	0.00	0.34	3.11	846.20	−0.10
r211	2	847.45	2	3.09	20.62	2	2.21	−1.41	3.53	827.89	−2.31
rc101	16	1726.91	16	0.00	8.05	16	0.24	0.47	1.45	1726.91	0.00
rc102	14	1659.53	14	0.00	8.57	15	−6.32	−6.15	1.88	1552.08	−6.47
rc103	13	1351.15	13	1.35	8.89	13	0.18	0.02	2.46	1350.09	−0.08
rc104	11	1229.82	11	0.00	8.79	11	1.58	−0.05	2.86	1227.25	−0.21
rc105	14	1475.31	14	0.23	7.88	14	0.55	0.62	1.48	1475.31	0.00
rc106	13	1436.61	13	0.00	7.63	13	0.25	0.17	1.66	1427.21	−0.65
rc107	12	1275.89	12	0.60	8.02	12	0.00	0.04	2.06	1274.89	−0.08
rc108	11	1204.87	11	0.00	7.68	11	2.82	−0.35	2.61	1197.83	−0.58
rc201	4	1444.94	4	1.34	23.02	4	0.16	0.13	1.77	1444.94	0.00
rc202	3	1412.91	3	1.71	25.12	3	0.00	0.45	3.24	1410.74	−0.15
rc203	3	1073.98	3	1.00	26.12	3	0.40	−0.01	3.88	1055.19	−1.75
rc204	3	885.35	3	1.96	26.45	3	0.44	0.80	4.08	884.80	−0.06
rc205	3	1282.58	3	0.00	23.42	3	3.05	0.56	3.19	1273.55	−0.70
rc206	3	1190.75	3	2.35	24.36	3	0.03	0.06	2.69	1188.63	−0.18
rc207	3	995.52	3	2.07	23.75	3	0.00	1.01	3.66	985.03	−1.05
rc208	3	837.82	3	1.20	24.64	3	0.03	−0.18	3.98	836.29	−0.18
CNV	440		442			441					
Avg. gap				0.59			0.36	−0.06			−0.52
Avg. gap (corr.)				0.73			0.49	0.11			−0.35
Avg. time				15.92			15.34		2.78		

6.4. Performance of our ALNS on instances of related problems

To assess the performance of the proposed ALNS, we conduct experiments on available benchmark instances for the related problems VRPTW and E-VRPTW, which are both special cases of the E-VRPTWMF. In the VRPTW, only ICCVs are available and no energy consumption is considered. In the E-VRPTW, only ECVs are available and instead of the energy consumption function described in Section 3, consumed energy is a linear function of the distance traveled. The following two sections present the detailed results.

6.4.1. Performance of ALNS on VRPTW instances

We use the well-known Solomon 100-customer instances as VRPTW benchmark set (Solomon, 1987). The set contains 56 instances, which are grouped according to the distribution of customer locations – random (R), clustered (C), random-clustered (RC) – and the considered scheduling horizon. Heuristic methods for VRPTW use a hierarchical objective (minimize number of vehicles first, minimize traveled distance second). Our ALNS is not designed to minimize vehicle routes but to minimize traveled distance with a given number of vehicles. Therefore, we set the initial number of routes for each instance to the best known number given in the literature. If ALNS is not able to determine a feasible solution with the given number of vehicles after β_{feas} iterations, the number of vehicles is increased by one.

As comparison method for our ALNS, we use the currently best-performing VRPTW method in the literature, the memetic algorithm proposed by Nagata, Bräysy, and Dullaert (2010) (NBD). Note that for all Solomon instances the solution found by NBD corresponds to the best-known solution in the literature (Vidal, Crainic, Gendreau, & Prins, 2013). Table 9 reports the best solution with respect to the vehicle number m and the distance traveled f out of 10 runs for our ALNS (out of 5 runs for NBD). Results are given as averages over the instance groups R1, R2, C1, C2, RC1 and RC2. Finally, the cumulative number of vehicles (CNV) and the cumulative traveled distance (CTD) over all instances are reported. ALNS achieves a gap to NBD of 0 percent concerning the CNV and a gap of 0.37 percent with regard to the CTD. Although a direct comparison of run-times between different platforms (using different operating systems, programming languages, and compilers) is always difficult, we state that ALNS has an average computation time of 87 seconds per run compared to 300 seconds of NBD. These results prove the capability of ALNS to find high-quality VRPTW solutions in fast computation times. This is a prerequisite to be able to address the computationally demanding E-VRPTWMF instances.

6.4.2. Performance of ALNS on E-VRPTW instances

Experiments for the E-VRPTW are conducted on the instance set presented in Schneider, Stenger, and Goeke (2014), where also a detailed description of the instances can be found. The set contains 56 instances which are based on the Solomon VRPTW instances but additionally contain 21 recharging stations. The vehicle fleet consists solely of ECVs with a limited battery capacity, whose energy consumption depends linearly on the distance traveled. Recharging times at stations are assumed a linear function of the required charge. The objective of E-VRPTW is hierarchical like in the VRPTW and we use the numbers of vehicles reported in Schneider, Stenger, and Goeke (2014) as start values for our ALNS. As comparison method, we use the ALNS of Hiermann, Puchinger, and Hartl (2014) (HPH), and the VNS/TS hybrid of Schneider, Stenger, and Goeke (2014) (SSG).

In Table 10, for each instance, we provide the previous best-known solution (BKS) as reported in Hiermann et al. (2014); Schneider, Stenger, and Goeke (2014) in terms of number of vehicles m and traveled distance f . For the algorithms HPH, SSG, and our ALNS, we report the best result out of 10 runs: the vehicle number m and the

percentage gap of the traveled distance to the BKS (Δ_{best}). Since our ALNS uses the same vehicle numbers as reported in Schneider, Stenger, and Goeke (2014), it is only reported once for SSG. Moreover, for HPH and our ALNS, we additionally provide the average run-time for each instance, for SSG only the average run-time over all instances is given. A precise comparison of run-times is only possible for ALNS and SSG because both algorithms were run on the same platform. However, although HPH was run on a different platform, we think that a rough comparison of run-times may be valid since all algorithms were run on modern computers with comparable computing power.

Column ALNS lists the overall best results that we found during the testing of our ALNS. The best solution for each instance is marked in bold. Moreover, we mark in italics those results for which the number of vehicles used by HPH and SSG/ALNS differ. Finally, the cumulative number of vehicles (CNV), the averages over the gaps and the average run-times in minutes are reported at the bottom of the table. The average gap is the average over all instances, the corrected average gap only includes the instances for which all algorithms use the same number of vehicles. We deem the latter measure more important due to the influence of the number of vehicles on the traveled distance. It may be quite easy to find a better distance when using one more vehicle, but this distorts the comparison between the algorithms with regard to traveled distance.

The results show that our ALNS outperforms HPH and SSG. In comparison to SSG, ALNS finds the same CNV and improves the average gap of the traveled distance from 0.49 percent to 0.11 percent. In comparison to HPH, ALNS is able to reduce the CNV by one vehicle and to improve the distance gap by 0.62 percent. The average run-time decreases from about 15 minutes for SSG and HPH to less than 3 minutes. Considering the solutions found during the overall testing, our ALNS is able to improve the previous BKS for 34 out of the 56 instances, for 20 instances the previous BKS is matched. On average, the previous BKS is improved by 0.35 percent.

7. Conclusion

We introduce the E-VRPTWMF, a VRP to optimize the routing of a mixed vehicle fleet consisting of ICCVs and ECVs. Contrary to existing VRPs for ECVs which assume energy consumption to be a linear function of traveled distance, we utilize a realistic energy consumption model that incorporates speed, gradient and load distribution. This is highly relevant in the context of ECVs, as energy consumption determines the maximal driving range of ECVs and the recharging times at stations. To address the problem, we develop an ALNS algorithm that is enhanced by an LS for intensification.

In numerical studies on newly designed E-VRPTWMF test instances, we find that consideration of the actual load strongly improves the quality of the generated solutions in comparison to solutions that are generated based on load estimates. Moreover, we find that a large number of solutions that are generated with “optimistic” load estimates are actually infeasible due to battery capacity or time window violations. We further show that our ALNS works effectively with all of the investigated cost functions and that the traditional objective of minimizing traveled distance fails to produce high-quality solutions if routing costs including fuel, labor and battery depreciation are considered. The choice of objective function additionally has a strong influence on the level of usage of the ECVs in the fleet. Finally, the performance of the developed algorithm is proven on benchmark instances of related problems: The ALNS provides convincing results in moderate run-times on the well-known Solomon VRPTW benchmark. On E-VRPTW instances, the ALNS outperforms all comparison methods with respect to both solution quality and run-time.

There are several interesting avenues for future research. From a modeling perspective, the real-world processes related to ECVs should be modeled in a more realistic fashion, see our discussion

at the end of [Section 4](#) on how to integrate topology and acceleration processes. Another important point in this respect is a more realistic modeling of the recharging processes. From the computational viewpoint, the development of exact solution methods for solving basic electric VRPs, and the design of efficient metaheuristics for solving rich variants featuring realistic energy consumption and recharging models as well as relevant side constraints of practical routing problems are both interesting topics.

Appendix A. Overview of complete results on E-VRPTWMF

To enable researchers that address the E-VRPTWMF in the future to compare the performance of their algorithms with ours, we provide results on the complete set of E-VRPTWMF instances introduced in [Section 6.2](#) (see [Table 11](#)). The reported results are based on the best of 10 ALNS runs. As objective function, we used the minimization of traveled distance f_d (in kilometers).

Table 11

Overview of results for the complete E-VRPTWMF instance set. Results for the best of 10 ALNS runs under the objective of distance minimization are reported. We provide the number of ICCVs (m_{IC}), the number of ECVs (m_E), the distance (f_d) and the average run-time (given in column t in minutes).

Inst.	m_{IC}	m_E	f_d	t	Inst.	m_{IC}	m_E	f_d	t	Inst.	m_{IC}	m_E	f_d	t
E-UK10_01	2	1	408.13	0.03	E-UK25_01	3	1	672.89	0.59	E-UK100_01	14	7	2820.29	7.02
E-UK10_02	2	1	527.13	0.02	E-UK25_02	4	2	812.97	0.22	E-UK100_02	13	6	2668.59	6.74
E-UK10_03	2	1	493.05	0.02	E-UK25_03	3	1	467.84	0.48	E-UK100_03	13	6	2476.95	7.55
E-UK10_04	2	1	495.66	0.02	E-UK25_04	3	1	611.53	0.34	E-UK100_04	14	7	2351.57	8.72
E-UK10_05	2	1	454.86	0.03	E-UK25_05	4	2	790.33	0.22	E-UK100_05	14	7	2281.46	6.62
E-UK10_06	2	1	595.80	0.02	E-UK25_06	4	2	715.26	0.25	E-UK100_06	14	7	2713.13	6.02
E-UK10_07	2	1	510.30	0.02	E-UK25_07	3	1	795.60	0.24	E-UK100_07	12	6	2412.81	6.80
E-UK10_08	2	1	581.21	0.02	E-UK25_08	3	1	874.11	0.18	E-UK100_08	13	6	2546.89	6.44
E-UK10_09	2	1	451.99	0.02	E-UK25_09	4	2	679.11	0.22	E-UK100_09	13	6	2191.43	6.71
E-UK10_10	2	1	569.74	0.02	E-UK25_10	4	2	895.39	0.30	E-UK100_10	12	6	2473.94	6.75
E-UK10_11	2	1	761.50	0.02	E-UK25_11	4	2	887.59	0.53	E-UK100_11	15	7	2736.90	6.11
E-UK10_12	2	1	472.83	0.07	E-UK25_12	4	2	1034.18	0.19	E-UK100_12	12	6	2353.43	6.84
E-UK10_13	2	1	524.57	0.02	E-UK25_13	4	2	511.47	0.48	E-UK100_13	13	6	2614.28	6.48
E-UK10_14	2	1	397.75	0.01	E-UK25_14	4	2	927.99	0.21	E-UK100_14	14	7	2902.97	6.16
E-UK10_15	2	1	291.37	0.12	E-UK25_15	3	1	879.95	0.17	E-UK100_15	15	7	2989.12	6.00
E-UK10_16	2	1	443.61	0.03	E-UK25_16	4	2	818.78	0.24	E-UK100_16	12	6	2200.01	6.43
E-UK10_17	2	1	387.52	0.06	E-UK25_17	4	2	1277.86	0.17	E-UK100_17	15	7	2907.88	6.28
E-UK10_18	2	1	394.53	0.03	E-UK25_18	3	1	942.12	0.18	E-UK100_18	13	6	2487.61	7.69
E-UK10_19	2	1	414.46	0.06	E-UK25_19	4	2	1068.07	0.19	E-UK100_19	13	6	2262.53	6.67
E-UK10_20	2	1	416.23	0.03	E-UK25_20	4	2	825.93	0.16	E-UK100_20	14	7	2899.56	6.06
E-UK15_01	2	0	709.01	0.03	E-UK50_01	7	3	1357.86	1.05	E-UK150_01	20	10	3046.28	17.67
E-UK15_02	2	1	517.16	0.06	E-UK50_02	7	3	1406.41	1.18	E-UK150_02	20	10	3831.61	12.18
E-UK15_03	3	1	751.86	0.08	E-UK50_03	7	3	1448.90	2.20	E-UK150_03	19	9	3132.22	15.38
E-UK15_04	3	1	757.34	0.08	E-UK50_04	8	4	1764.76	1.25	E-UK150_04	21	10	3703.23	10.88
E-UK15_05	2	1	891.94	0.06	E-UK50_05	6	3	1568.67	1.16	E-UK150_05	20	10	3276.74	12.06
E-UK15_06	3	1	551.15	0.13	E-UK50_06	8	4	1283.86	2.05	E-UK150_06	21	10	3195.89	14.89
E-UK15_07	3	1	604.83	0.10	E-UK50_07	7	3	1193.06	1.74	E-UK150_07	21	10	3856.46	10.89
E-UK15_08	2	1	431.17	0.07	E-UK50_08	7	3	1271.94	1.10	E-UK150_08	20	10	3447.95	11.63
E-UK15_09	3	1	637.57	0.07	E-UK50_09	7	3	1611.98	0.98	E-UK150_09	20	10	3789.48	10.51
E-UK15_10	2	1	568.48	0.08	E-UK50_10	7	3	1564.27	1.95	E-UK150_10	20	10	3616.08	10.24
E-UK15_11	2	0	644.95	0.08	E-UK50_11	7	3	1473.79	1.15	E-UK150_11	20	10	3818.09	11.33
E-UK15_12	3	1	779.42	0.05	E-UK50_12	7	3	1318.96	1.45	E-UK150_12	21	10	4018.53	10.47
E-UK15_13	3	1	593.14	0.08	E-UK50_13	7	3	1324.20	0.91	E-UK150_13	19	9	3686.79	10.73
E-UK15_14	3	1	846.94	0.04	E-UK50_14	7	3	1542.86	1.73	E-UK150_14	20	10	3729.52	10.42
E-UK15_15	2	1	584.35	0.06	E-UK50_15	6	3	1373.09	1.26	E-UK150_15	19	9	3107.91	11.53
E-UK15_16	2	1	556.12	0.06	E-UK50_16	7	3	1322.54	1.63	E-UK150_16	20	10	3772.39	12.16
E-UK15_17	3	1	693.26	0.04	E-UK50_17	7	3	938.50	1.01	E-UK150_17	20	10	3805.71	11.24
E-UK15_18	3	1	825.43	0.05	E-UK50_18	8	4	1601.10	1.05	E-UK150_18	20	10	3739.99	11.78
E-UK15_19	2	1	376.11	0.18	E-UK50_19	7	3	1347.43	1.26	E-UK150_19	20	10	4161.73	10.78
E-UK15_20	3	1	469.97	0.16	E-UK50_20	7	3	1573.49	1.24	E-UK150_20	20	10	4051.19	11.47
E-UK20_01	3	1	785.47	0.13	E-UK75_01	11	5	2227.65	2.73	E-UK200_01	28	14	4616.77	17.11
E-UK20_02	3	1	881.53	0.13	E-UK75_02	11	5	1803.68	3.00	E-UK200_02	24	12	4478.89	17.44
E-UK20_03	3	1	456.73	0.25	E-UK75_03	10	5	1965.21	2.91	E-UK200_03	27	13	4420.04	16.07
E-UK20_04	3	1	800.39	0.18	E-UK75_04	11	5	1689.82	3.72	E-UK200_04	26	13	4175.85	16.45
E-UK20_05	3	1	718.34	0.11	E-UK75_05	10	5	2009.95	2.98	E-UK200_05	27	13	4868.83	15.37
E-UK20_06	3	1	839.43	0.15	E-UK75_06	11	5	2085.90	2.83	E-UK200_06	27	13	4044.02	16.26
E-UK20_07	3	1	522.23	0.46	E-UK75_07	11	5	2201.08	2.49	E-UK200_07	27	13	4334.31	16.40
E-UK20_08	3	1	700.59	0.14	E-UK75_08	10	5	2334.70	3.52	E-UK200_08	27	13	4679.70	24.72
E-UK20_09	3	1	796.50	0.09	E-UK75_09	10	5	2155.98	2.84	E-UK200_09	25	12	4026.65	16.91
E-UK20_10	3	1	696.79	0.35	E-UK75_10	11	5	2232.32	2.97	E-UK200_10	28	14	4903.01	15.23
E-UK20_11	3	1	932.86	0.12	E-UK75_11	10	5	1427.25	2.79	E-UK200_11	27	13	4130.06	16.12
E-UK20_12	3	1	794.58	0.12	E-UK75_12	10	5	1971.57	2.97	E-UK200_12	25	12	4845.70	15.86
E-UK20_13	3	1	764.83	0.11	E-UK75_13	10	5	2263.37	3.08	E-UK200_13	25	12	4847.07	16.45
E-UK20_14	4	2	1045.96	0.11	E-UK75_14	10	5	2146.74	3.18	E-UK200_14	27	13	4406.88	16.12
E-UK20_15	3	1	788.34	0.11	E-UK75_15	10	5	2346.76	2.65	E-UK200_15	25	12	4766.98	16.04
E-UK20_16	3	1	829.14	0.09	E-UK75_16	10	5	2132.85	2.73	E-UK200_16	27	13	4520.69	15.98
E-UK20_17	3	1	873.35	0.14	E-UK75_17	11	5	2057.84	2.60	E-UK200_17	26	13	5006.00	14.74
E-UK20_18	3	1	838.96	0.10	E-UK75_18	10	5	1882.89	3.09	E-UK200_18	27	13	4453.07	20.28
E-UK20_19	3	1	812.21	0.11	E-UK75_19	10	5	1865.06	2.22	E-UK200_19	25	12	3999.15	15.30
E-UK20_20	3	1	873.35	0.11	E-UK75_20	11	5	2028.89	2.21	E-UK200_20	27	13	4825.55	15.44

References

- Artmeier, A., Haselmayr, J., Leucker, M., & Sachenbacher, M. (2010). The shortest path problem revisited: Optimal routing for electric vehicles. In *KI 2010: Advances in artificial intelligence*. Lecture Notes in Computer Science (Vol. 6359, pp. 309–316). Springer, Berlin.
- Barco, J., Guerra, A., Muñoz, L., & Quijano, N. (2013). *Optimal routing and scheduling of charge for electric vehicles: Case study* (Working paper). URL: <http://arxiv.org/pdf/1310.0145v1.pdf>.
- Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8), 1232–1250.
- Bousonville, T., Hartmann, A., Melo, T., & Kopfer, H. (2011). Vehicle routing and refueling: the impact of price variations on tour length. In *Logistikmanagement - Herausforderungen, Chancen und Lösungen (Proceedings of LM11)* (Vol. 2, pp. 83–101). Conrad, G. R., & Figliozzi, A. M. (2011). The recharging vehicle routing problem. In T. Doolen, & E. Van Aken (Eds.), *Proceedings of the 2011 industrial engineering research conference*. USA: Reno.
- Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52(8), 928–936.
- Crainic, T. G., Gendreau, M., Soriano, P., & Toulouse, M. (1993). A tabu search procedure for multicommodity location/allocation with balancing requirements. *Annals of Operations Research*, 41(4), 359–383.
- Crevier, B., Cordeau, J.-F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2), 756–773.
- Davis, B. A., & Figliozzi, M. A. (2013). A methodology to evaluate the competitiveness of electric delivery trucks. *Transportation Research Part E: Logistics and Transportation Review*, 49(1), 8–23.
- Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2013). *An adaptive large neighborhood search heuristic for a multi-period vehicle routing problem* (Technical report 2013-67). CIRRELT, Canada.
- Dekker, R., Bloemhof, J., & Mallidis, I. (2012). Operations research for green logistics: An overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3), 671–679.
- Demir, E., Bektaş, T., & Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5), 347–357.
- Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2), 346–359.
- Demir, E., Bektaş, T., & Laporte, G. (2014a). The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3), 464–478.
- Demir, E., Bektaş, T., & Laporte, G. (2014b). A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3), 775–793.
- Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2014). Exact algorithms for electric vehicle-routing problems with time windows. *Les Cahiers du GERAD G-2014-110*. GERAD, Canada.
- Erdogan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 100–114.
- Felipe, A., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71, 111–128.
- Feng, W., & Figliozzi, M. A. (2013). An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. *Transportation Research Part C: Emerging Technologies*, 26, 135–145.
- Feo, T. A., & Resende, M. G. C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2), 67–71.
- Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10), 1276–1290.
- Gendreau, M., & Potvin, J.-Y. (2010). Tabu search. In M. Gendreau, & J.-Y. Potvin (Eds.), *Handbook of metaheuristics*. International series in operations research & management science (Vol. 146, pp. 41–59). Springer.
- Guzzella, L., & Amstutz, A. (2005). The QSS toolbox manual. URL: <http://www.idsc.ethz.ch/Downloads/DownloadFiles/qss>. Accessed 10.05.13.
- He, F., Wu, D., Yin, Y., & Guan, Y. (2013). Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 47, 87–101.
- Heineken International (2014). *Case studies: Europe's largest electric truck will drive down emissions*. (Sustainability report 2013). URL: <http://sustainabilityreport.heineken.com/Reducing-CO2-emissions/Case-studies/Europes-largest-electric-truck-will-drive-down-emissions/index.htm>. Accessed 17.07.14.
- Hemmelmayr, C. V., Doerner, F. K., Hartl, F. R., & Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, 19(2), 129–156.
- Hemmelmayr, V. C., Cordeau, J.-F., & Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12), 3215–3228.
- Hiermann, G., Puchinger, J., & Hartl, R. F. (2014). *The electric fleet size and mix vehicle routing problem with time windows and recharging stations* (Working paper). URL: http://prolog.univie.ac.at/research/publications/downloads/Hie_2014_638.pdf. Accessed 17.07.14.
- Hoke, A., Brissette, A., Maksimovic, D., Pratt, A., & Smith, K. (2011). Electric vehicle charge optimization including effects of lithium-ion battery degradation. In *Vehicle power and propulsion conference (vppc)* (pp. 1–8). Chicago, USA: IEEE.
- International Energy Agency (2012). *EV city casebook: A look at the global electric vehicle movement*. URL: <http://www.iea.org/publications/freepublications/publication/EVCityCasebook.pdf>. Accessed 17.07.14.
- International Energy Agency (2013). *EV outlook: Understanding the electric vehicle landscape to 2020*. URL: http://www.iea.org/publications/freepublications/publication/GlobalEVOutlook_2013.pdf. Accessed 17.07.14.
- Jabali, O., Van Woensel, T., & de Kok, G. A. (2012). Analysis of travel times and CO₂ emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6), 1060–1074.
- Kindervater, G., & Savelsbergh, M. (1997). Vehicle routing: Handling edge exchanges. In E. Aarts, & J. K. Lenstra (Eds.), *Local search in combinatorial optimization* (pp. 337–360). John Wiley & Sons.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
- Kleindorfer, P. R., Neboian, A., Roset, A., & Spinler, S. (2012). Fleet renewal with electric vehicles at La Poste. *Interfaces*, 42(5), 465–477.
- Kopfer, H. W., Schönberger, J., & Kopfer, H. (2014). Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. *Flexible Services and Manufacturing Journal*, 26(1–2), 221–248.
- Laporte, G., Nobert, Y., & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5), 1050–1073.
- Mak, H.-Y., Rong, Y., & Shen, Z.-J. M. (2013). Infrastructure planning for electric vehicles with battery swapping. *Management Science*, 59(7), 1557–1575.
- Marra, F., Yang, G. Y., Traholt, C., Larsen, E., Rasmussen, C. N., & Shi, Y. (2012). Demand profile study of battery electric vehicle under different charging options. In *Power and energy society general meeting, 2012* (pp. 1–7). San Diego, USA: IEEE.
- Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2014). *A modified multi-space sampling heuristic for the green vehicle routing problem* (Technical report LARIS-EA 7315). Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Université d'Angers.
- Nagata, Y., Bräysy, O., & Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4), 724–737.
- National Renewable Energy Laboratory (2014). *Project startup: Evaluating the performance of Frito Lay's electric delivery trucks*. URL: <http://www.nrel.gov/docs/fy14osti/61455.pdf>. Accessed 23.07.14.
- Nie, Y. M., & Ghamami, M. (2013). A corridor-centric approach to planning electric vehicle charging infrastructure. *Transportation Research Part B: Methodological*, 57, 172–190.
- Nie, Y. M., & Li, Q. (2013). An eco-routing model considering microscopic vehicle operating conditions. *Transportation Research Part B: Methodological*, 55, 154–170.
- Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *The Journal of the Operational Research Society*, 46(12), 1433–1446.
- Preis, H., Frank, S., & Nachtigall, K. (2014). Energy-optimized routing of electric vehicles in urban delivery systems. In S. Helber, M. Bretnier, D. Rösch, C. Schön, J.-M. Graf von der Schulenburg, P. Sibbertsen, M. Steinbach, S. Weber, & A. Wolter (Eds.), *Operations research proceedings 2012 – Selected papers of the international annual conference of the german operations research society (GOR)* (pp. 583–588). Springer.
- Ropke, S., & Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472.
- Ropke, S., & Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3), 750–775.
- Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2), 146–154.
- Sbihi, A., & Eglese, R. W. (2010). Combinatorial optimization and green logistics. *Annals of Operations Research*, 175(1), 159–175.
- Schneider, M., Sand, B., & Stenger, A. (2013). A note on the time travel approach for handling time windows in vehicle routing problems. *Computers & Operations Research*, 40(10), 2564–2568.
- Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4), 500–520.
- Schneider, M., Stenger, A., & Hof, J. (2014). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*. doi:10.1007/s00291-014-0376-5.
- Shaw, P. (1997). *A new local search algorithm providing high quality solutions to vehicle routing problems* (Technical report). Glasgow, Scotland: APES Group, Department of Computer Science, University of Strathclyde.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher, & J.-F. Puget (Eds.), *Principles and practice of constraint programming – CP'98*. Lecture notes in computer science (Vol. 1520, pp. 417–431). London, UK: Springer.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254–265.
- Suzuki, Y. (2012). A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands. *Decision Support Systems*, 54(1), 758–767.
- Tesla Motors, Inc. (2014). *Supercharger*. URL: <http://www.teslamotors.com/supercharger>. Accessed 07.07.14.
- Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4), 333–346.

- Toth, P., & Vigo, D. (Eds.), (2014). *Vehicle routing: Problems, methods, and applications*, vol. 18 (pp. 119–159). MOS-SIAM Series on Optimization (2nd ed.). Philadelphia, USA: SIAM.
- van Keulen, T., de Jager, B., Serrarens, A., & Steinbuch, M. (2010). Optimal energy management in hybrid electric trucks using route information. *Oil and Gas Science and Technology*, 65(1), 103–113.
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1), 475–489.
- Wang, Y.-W., & Lin, C.-C. (2013). Locating multiple types of recharging stations for battery-powered electric vehicle transport. *Transportation Research Part E: Logistics and Transportation Review*, 58, 76–87.
- Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7), 1419–1431.
- Zhang, J., Tang, J., & Fung, R. Y. K. (2011). A scatter search for multi-depot vehicle routing problem with weight-related cost. *Asia-Pacific Journal of Operational Research*, 28(3), 323–348.