



# A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows

Sina Rastani<sup>1,2,3</sup>  · Bülent Çatay<sup>2,3</sup>

Accepted: 9 September 2021 / Published online: 8 November 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Range anxiety of electric vehicles (EVs) still poses a major barrier in their adoption in the logistics operations despite the advancements in the battery technology. The need for recharging the battery during the day brings additional complexities to the operational planning of commercial EVs in last mile deliveries. The driving range of an EV may vary according to different factors including ambient temperature, weight, speed, acceleration/deceleration, and the road profile. In this study, we revisit the well-known electric vehicle routing problem with time windows by taking into account the weight of the load carried. Cargo weight may play a crucial role in the operational efficiency of the EVs since it may affect the energy consumption significantly. We first present two alternative mathematical formulations of the problem and test their performances on small-size instances that can be solved using a commercial solver. Next, we develop a matheuristic approach that integrates an optimal repair procedure in the large neighbourhood search method and validate its performance. Then, we present an extensive numerical study to investigate the influence of load on the routing decisions. Our results show that cargo weight may create substantial changes in the route plans and fleet size, and neglecting it may cause severe disruptions in service and increase the costs.

**Keywords** Electric vehicle routing · Time windows · Load-dependent · Energy consumption

## 1 Introduction

There has been a growing demand in the developed countries for reducing the adverse effects of transport activities on nature and human life, and liberating city centers from emissions. Governments have started encouraging the adoption of alternative fuel vehicles (AFVs) by offering new incentives (Wyman, 2018) or by restricting the sales of internal combustion

---

✉ Sina Rastani  
srastani@sabanciuniv.edu

<sup>1</sup> Industrial and Systems Engineering Department, Yeditepe University, Istanbul, Turkey

<sup>2</sup> Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

<sup>3</sup> Smart Mobility and Logistics Lab, Sabanci University, Istanbul, Turkey

engine vehicles (ICEVs) to eliminate them in the future (Edelstein, 2016; Khan, 2016). Consequently, the utilization of AFVs has become an emerging trend in the transport sector. The electric vehicles (EVs), in particular, have attracted great attention in logistics operations as they cut dependency on fossil fuels, hence lessen the air pollution. In addition to zero-emission and noise, the main operational and financial advantages of EVs over ICEVs are low electricity prices compared to fossil fuels, high efficiency of the electric motor and transmission system, regenerative braking, and reduced maintenance costs (Wu et al., 2015). On the other hand, operating EVs in logistics poses several challenges such as limited battery capacity and short driving range, scarcity of recharging stations, and long recharging times (Keskin & Çatay, 2018). Because of these additional complexities, route planning of EVs has recently gained popularity in the vehicle routing problem (VRP) literature, coining the name the electric VRP (EVRP).

EVRP is an extension of VRP where an EV fleet is utilized instead of ICEV. Conrad and Figliozzi (2011) introduced an EVRP where the EVs are allowed to recharge their battery at selected customer locations during the delivery service. The recharge duration is assumed constant. Schneider et al. (2014) addressed EVRP with time window (EVRPTW) by relaxing this assumption but adopting a full-recharge strategy. The full-recharge assumption was later relaxed by Bruglieri et al. (2015) and Keskin and Çatay (2016) and partial recharges at any quantity were allowed, which constitutes the realistic and practical case.

In this paper, we address the load-dependent EVRPTW (LD-EVRPTW) with partial recharges by taking into consideration the energy consumption related to the weight of the cargo carried on the vehicle. LD-EVRPTW is expected to offer more realistic route plans as compared to other routing models for companies that carry heavy goods such as appliances, machinery, hardware, liquids (Zachariadis et al., 2015), which comprises the main motivation of our study. Similar to the literature on EVRPTW, we adopt a hierarchical objective function where the primary objective is to minimize the fleet size whereas the secondary objective is to minimize the total energy consumption. We model this problem using two different mathematical programming formulations, one is linear whereas the other is non-linear. Since the large-size problems are not tractable, we propose a large neighbourhood search (LNS)-based matheuristic approach to solve them effectively. Our LNS method benefits from an optimal repair procedure which employs a commercial solver to solve the non-linear model. Finally, we analyse the influence of the cargo load on the routing decisions through an extensive numerical study.

The contributions of this study can be summarized as follows:

- We extend EVRPTW by considering the load weight which varies throughout the journey of the EV and introduce the LD-EVRPTW.
- We present linear and non-linear mathematical programming formulations of the problem, and show that the non-linear model performs better in terms of solution quality and run time.
- We develop an effective LNS-based matheuristic method to solve the LD-EVRPTW by adapting the destroy and repair operators from the literature and introducing new ones to tackle the more complex structure of the problem including an optimal repair mechanism.
- We validate the performance of the proposed method using the benchmark instances and show that the route plans, hence the operational costs, may change substantially in the load-dependent case.

The remainder of the paper is organized as follows: Sect. 2 reviews the related literature. Section 3 introduces the problem, formulates two alternative mathematical programming

models, and tests their effectiveness. Section 4 presents the proposed LNS-based matheuristic method. Section 5 describes the experimental study and discusses the results. Finally, concluding remarks are provided in Sect. 6.

## 2 Related literature

EVRP with time window (EVRPTW) was presented by Schneider et al. (2014) where a full-recharge strategy was adopted. The authors developed the mathematical programming formulation of the problem and proposed a hybrid variable neighbourhood search (VNS) and tabu search (TS) algorithm to solve it. Different variants of EVRP and EVRPTW were addressed in several studies including the cases of partial recharge (Bruglieri et al., 2015; Keskin & Çatay, 2016), mixed fleet (Sassi et al., 2015; Goeke & Schneider, 2015; Hiermann et al., 2016; Kopfer & Vornhusen, 2019; Hiermann et al., 2019), location routing (Li-Ying & Yuan-Bin, 2015; Paz et al., 2018; Schiffer & Walther, 2017; Yang & Sun, 2015), different type of chargers at the stations (Çatay & Keskin, 2017; Felipe et al., 2014; Keskin & Çatay, 2018; Li-Ying & Yuan-Bin, 2015; Sassi, et al., 2015), non-linear charging function (Froger et al., 2019; Montoya et al., 2017), battery swapping (Hof et al., 2017; Jie, et al., 2019; Liao et al., 2016; Masmoudi et al., 2018; Paz et al., 2018; Wang et al., 2018; Yang & Sun, 2015), wireless charging systems along the roads (Li et al., 2018), flexible time windows (Taş, 2020), two-echelons (Breunig et al., 2019; Jie et al., 2019) and pick-up and delivery problem (Grandinetti et al., 2016; Madankumar & Rajendran, 2018). Desaulniers et al. (2016) also studied EVRPTW and proposed a branch-price-and-cut algorithm to solve four different recharging strategies. Some recent studies have dealt with the availability of recharging stations and queueing for recharging service (Froger et al., 2017; Keskin et al., 2019; Kullman et al., 2019). A comprehensive review of the EV technology and survey of the EVRP variants may be found in Pelletier et al. (2016), Pelletier et al. (2017), and Erdelić and Carić (2019).

Apart from distance traveled by the vehicle, there are other factors that affect the energy consumption on the road, such as the weight of the vehicle, its speed, auxiliary equipment (internal factors) as well as road gradient and ambient temperature (external factors). These factors have been often neglected within the VRP literature either because they make the problem too complex to solve or the driving range is not an issue as the vehicles will easily refuel at a close-by gas station. To the best of our knowledge, Kara et al. (2007) is the first study that proposed a load-dependent cost function for the capacitated VRP (CVRP), which is calculated by multiplying the distance of the arc with the load of the vehicle when traversing it. Xiao et al. (2012) considered the effect of cargo load on the fuel consumption within the context of fuel consumption CVRP (FCVRP) and developed a simulated annealing (SA) algorithm with a hybrid exchange rule to solve the problem. Zachariadis et al. (2015) extended the load-dependent VRP (LDVRP) by considering simultaneous pick-ups and deliveries and proposed a local-search algorithm to solve it.

Fukasawa et al. (2016) proposed two mixed-integer linear programming formulations for the LDVRP and solved them using branch-and-cut and branch-cut-and-price algorithms. Corberán et al. (2018) considered the effect of load weight on the chinese postman problem (CPP). They introduced two formulations for the problem and solved large-size instances using iterated local search (ILS) and VNS algorithms. Liu and Jiang (2019) addressed the LDVRP with time windows (LDVRPTW) and proposed a constraint relaxation-based algorithm.

The internal and external factors may play a more important role in the operational efficiency of the EVs as they can increase their energy consumption considerably, causing frequent visits to recharging stations which are scarce. Several articles investigated the influence of ambient temperature (Liu et al., 2018; Neubauer & Wood, 2014; Yuksel & Michalek, 2015; Yuksel et al., 2016), driver behavior (Guo et al., 2020; Neubauer & Wood, 2014; Wu et al., 2015), terrain (Asamer et al., 2016), speed (Morlock et al., 2019) on the EV energy consumption. However, the effects of these factors on routing decisions have received little attention in the VRP literature. Among them, Rastani et al. (2019) investigated the influence of ambient temperature on the fleet composition and route plans in the EVRPTW. Goeke and Schneider (2015) considered a mixed fleet of EVs and ICEVs, and formulated a nonlinear mixed integer model using the energy consumption model of Demir et al. (2012) that incorporates speed, road gradient, and load. This model is based on an augmented set of recharging stations, and determining the numbers of dummy stations requires additional computational effort. The authors developed an ALNS approach and analyzed three cost functions under different scenarios to provide managerial insights using instances adapted from the pollution routing problem (PRP) dataset of Demir et al. (2012).

In Table 1, we summarize the relevant literature by classifying the studies with respect to the main features of the problems that they addressed and solution methodologies that they proposed. Column “Load” reports whether the load effect is considered in the corresponding study. “Energy Cost Function” and “Objective Function” columns show the nature of the energy consumption and objective functions. The numbers given under the Objective Function column are explained in Table 2. “Recharger Technology” denotes the charging infrastructure in the stations, which includes single, multiple, or wireless charging technology stations (WCS) as well as battery swapping stations (BSS). “Recharge Amount” column indicates whether the batteries are fully or partially recharged. The following column presents the solution methodology, where A- and EVO- VNS, HC, DP, WSM, and CG stand for adaptive- and evolutionary-VNS, heuristic concentration, dynamic programming, weighted sum method, and column generation, respectively. The studies that only presented the solutions from a mixed-integer programming solver are reported as MIP. Finally, the last two columns show whether time windows and station location decisions are considered in the corresponding problem.

### 3 Problem description and model formulation

We tackle EVRPTW where a homogeneous fleet of EVs serve a set of customers with known demands, time windows, and service times. As opposed to previous studies which assume that the energy on the battery is consumed proportional to the distance traveled, we consider the additional energy consumption associated with cargo weight. The partial recharge strategy is adopted where the duration of recharge depends on the amount of energy transferred. Since it is a common practice in the real world to operate within the first phase of recharging where the energy transferred is a linear function of the recharge duration in order to prolong the battery life (Pelletier et al., 2017), we assume a linear charging function. Furthermore, we assume that the EV can be recharged at most once between two consecutive customers, which is the practical case in most last-mile operations. Without loss of generality, we consider a pick-up problem where the load of the EV increases during the journey as it visits the customers. Since EVs can be recharged overnight, they depart from the depot with full battery.

**Table 1** Overview of the related literature

Reference	Load	Energy Cost Function	Objective Function	Fleet composition		Recharge		Solution Method	TW	Location
				Homogeneous	Heterogeneous	Technology	Amount			
Breunig et al. (2019)		Linear	1, 2	EVs		Single	Full	LNS, Exact algorithm		
Bruglieri et al. (2015)		Linear	1, 2, 3, 4	EVs		Single	Partial	VNS branching	✓	
Çatay and Keskin (2017)		Linear	3	EVs		Multi	Partial	MIP	✓	
Conrad and Fighozzi (2011)		Linear	1, 2, 3, 6	EVs		Single	Full and Partial	Iterative construction, improvement heuristics		
Corberán et al. (2018)	✓	Nonlinear	2	ICEV				ILS, VNS		
Demir et al. (2012)	✓	Nonlinear	9, 14	ICEVs				ALNS	✓	
Desaulniers et al. (2016)		Linear	2	EVs		Single	Full and Partial	Branch-price-and-cut	✓	
Felipe et al. (2014)		Linear	2, 3	EVs		Multi	Partial	Several heuristics and SA	✓	
Froger et al. (2017)		Linear	6	EVs		Single	Partial	Two-stage metaheuristic		
Froger et al. (2019)		Linear	6	EVs		Single	Partial	MIP		
Fukasawa et al. (2016)	✓	Linear	2	ICEVs				Branch-cut and branch-cut-and-price algorithms		

Table 1 (continued)

Reference	Load	Energy Cost Function	Objective Function	Fleet composition		Recharge		Solution Method	TW	Location
				Homogeneous	Heterogeneous	Technology	Amount			
Goeke and Schneider (2015)	✓	Nonlinear	2, 3, 6, 7, 8	EVs, ICEVs	Single	Partial	ALNS	✓		
Grandinetti et al. (2016)		Linear	1, 2, 13	EVs	Single	Partial	WSM	✓		
Hiermann et al. (2016)		Linear	1, 2	EVs	Single	Full	Branch-price, and ALNS	✓		
Hiermann et al. (2019)		Linear	1, 3, 7	EVs, ICEVs, PHEVs	Single	Partial	GA, LNS with IP solver	✓		
Hof et al. (2017)		Linear	2, 5	EVs	BSS		AVNS		✓	
Jie et al. (2019)		Linear	2, 11, 12	EVs	BSS	Partial	Hybrid CG and ALNS			
Kara et al. (2007)	✓	Linear	2	ICEVs	Single	Partial	MIP	✓		
Keskin and Çatay (2016)		Linear	1, 2	EVs	Single	Partial	ALNS	✓		
Keskin and Çatay (2018)		Linear	1, 3	EVs	Multi	Partial	ALNS based matheuristic	✓		
Keskin et al. (2019)		Linear	1, 2, 13, 14	EVs	Single	Partial	ALNS based matheuristic	✓		
Kopfer and Vormhusen (2019)		Nonlinear	9	EVs, ICEVs	Single	Partial	MIP	✓		
Kullman et al. (2019)		Linear	6	EVs	Single	Partial	Several exact and heuristic methods			
Li et al. (2018)		Linear	3, 6	EVs	Single and WCS	Partial	MIP			
Liao et al. (2016)		Linear	6	EVs	BSS		DP			

Table 1 (continued)

Reference	Load	Energy Cost Function	Objective Function	Fleet composition		Recharge		Solution Method	TW	Location
				Homogeneous	Heterogeneous	Technology	Amount			
Liu and Jiang (2019)	✓	Nonlinear	2	ICEVs				Constraint Relaxation-based algorithm	✓	
Li-ying and Yuan-bin (2015)		Linear	3, 5, 6	EVs		Multi		Hybrid AVNS and TS	✓	✓
Madankumar and Rajendran (2018)		Linear	2	AFVs		Single	Full and Partial	MIP	✓	
Masmoudi et al. (2018)		Linear	2, 12		EVs	BSS		EVO-VNS	✓	
Montoya et al. (2017)		Linear	6	EVs		Multi	Partial	Hybrid ILS and HC		
Paz et al. (2018)		Linear	1, 2, 5	EVs		BSS	Full and Partial	MIP	✓	✓
Rastani et al. (2019)		Linear	1, 2	EVs		Single	Partial	ALNS	✓	
Sassi et al. (2015)		Linear	1, 2, 3		EVs, ICEVs	Multi	Partial	Several heuristics	✓	
Schiffer and Walther (2017)		Linear	1, 2, 5	EVs		Single	Full and Partial	MIP	✓	✓
Schiffer et al. (2018)		Linear	1, 2, 5, 15, 16	ICEVs		Single	Partial	ALNS, LS	✓	✓
Schneider et al. (2014)		Linear	1, 2	EVs		Single	Full	Hybrid VNS and TS	✓	
Taş (2020)		Linear	1, 2, 13	EVs		Single	Full	Column Generation	✓	
Wang et al. (2018)		Linear	10	EVs		BSS	Full	Heuristic procedures	✓	
Wen et al. (2016)		Linear	1, 2	EVs		Single	Partial	ALNS	✓	

Table 1 (continued)

Reference	Load	Energy Cost Function	Objective Function	Fleet composition		Recharge		Solution Method	TW	Location
				Homogeneous	Heterogeneous	Technology	Amount			
Xiao et al. (2012)	✓	Linear	1, 7	ICEVs				SA		
Yang and Sun (2015)		Linear	2, 5	EVs		BSS		Several heuristics		✓
Zachariadis et al. (2015)	✓	Linear	2	ICEVs				LS		
This Study	✓	Linear and Nonlinear	1, 2	EVs		Single	Partial	LNS based matheuristic	✓	



**Table 2** Classification of the objective functions in Table 1

1:	Vehicle cost	9:	Charging cost
2:	Total travel cost	10:	Profit of visits
3:	Recharging cost	11:	Operational costs
4:	Waiting cost	12:	Battery swapping cost
5:	Station installation cost	13:	Unit time penalty for violated time windows
6:	Total time cost	14:	Driver wage
7:	Fuel cost	15:	Fixed costs of building a freight replenishment facility
8:	Battery cost	16:	Fixed costs of building a combined facility

### 3.1 Energy consumption model

The energy consumption of an EV on the road depends on different factors such as its shape, mass, road gradient, acceleration, etc. The power demand of a vehicle can be obtained by using tractive power requirements placed on the vehicle at the wheels as follows (Demir et al., 2012):

$$F = Ma + Mgsin\theta + 0.5C_d\rho Av^2 + MgC_r cos\theta \quad (1)$$

$$P_{tract}(kW) = Fv/1000 \quad (2)$$

Equation (1) calculates the tractive effort  $F$  where  $M$  is the total weight of the vehicle, including its curb weight and the cargo weight ( $kg$ ),  $a$  is the acceleration ( $m/sec^2$ ),  $g$  is the gravitational constant,  $\theta$  is road gradient,  $C_d$  is the coefficient of aerodynamic drag,  $\rho$  is the air density in ( $kg/m^3$ ),  $A$  is the frontal area,  $v$  is the speed ( $m/s$ ), and  $C_r$  is the coefficient of rolling resistance. The tractive power requirement can be converted to second-by-second battery power output ( $kW$ ) as follows:

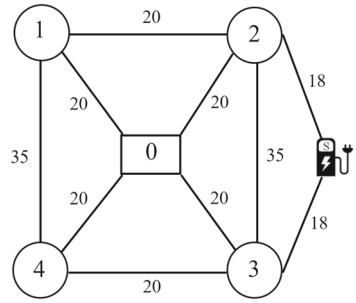
$$P = P_{tract}/\mu_{tf} + P_{acc} \quad (3)$$

where  $\mu_{tf}$  is the vehicle's drive train efficiency that includes the energy losses between electric motor and battery as well as the energy losses in transforming energy to the wheels.  $P_{acc}$  shows the power demand associated with the auxiliary equipment such as air conditioning, audio system, cabin lights, which are neglected in this study. Then, the energy consumption in ( $kWh/km$ ) can be calculated as follows:

$$E = P/v \quad (4)$$

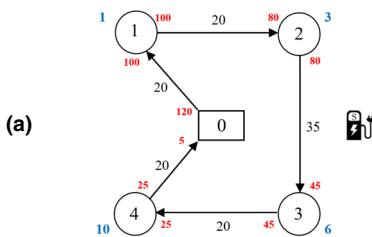
To demonstrate how cargo load may affect the routing decisions we present an illustrative example in Fig. 1. Our example consists of a network that includes a depot (node 0), one recharging station, and four customers whose demands have a weight equal to their ID number (e.g., customer 3 has a demand that weighs 3 units). The time windows are omitted for the sake of simplicity. The battery capacity of the EV is considered 120 units. The EV starts its trip from the depot empty and picks up the customers' demand by visiting them and then returns to the depot. The distances given next to the arcs are symmetric, and the speed of the EV is constant and equal to one (one unit distance is traveled in one unit of time). The consumption rate of an empty vehicle is assumed as one unit of energy per unit distance travelled, and the additional consumption corresponding to carrying one unit of cargo is

**Fig. 1** Illustrative example with one depot (charging station) and 4 customers

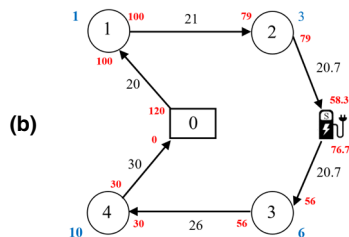


assumed to be 0.05 unit of energy per unit distance. The recharging rate is 2 time units per unit of energy.

Figure 2 illustrates the optimal solutions for the load-independent and load-dependent cases. The battery states of charge (SoCs) of the EV arriving at and departing from a node are indicated in red and the cargo loads at departure from the customers are shown in blue. The energy consumptions along the arcs are provided next to the corresponding arc. Figure 2a depicts the optimal solution for the load-independent case. In this case, since the effect of the cargo load is neglected, the energy consumption on an arc is equal to its distance as the energy consumption per unit distance is equal to one. We observe that the EV can serve all the customers without needing any recharge throughout its tour. Figure 2b shows the optimal solution for the load-dependent case. In this case, the EV needs to visit the recharging station after serving customer 2, the amount of energy recharged is 18.4 units and its duration is  $2 \times 18.4 = 36.8$ . Total energy consumption is 138.4 whereas it is 115 if the load is neglected. Note that neglecting the load did not only underestimate the cost in this case but it provided a route on which the EV would run out of battery. Note also that in the existence of time windows, one EV may not be able to serve all customers in a single tour and an additional vehicle may be needed.



Distance = Energy Consumption = Time = 115  
Energy Recharged en-route = 0



Distance = 116, Energy Consumption = 138.4  
Time = 152.8, Energy Recharged en-route = 18.4

**Fig. 2** Optimal routes for the illustrative example: (a) load-independent case, (b) load-dependent case

### 3.2 Mathematical models

In this section we present two alternative mathematical formulations of the LD-EVRPTW: Model 1 is a mixed-integer linear program whereas Model 2 involves a non-linear objective function with a set of linear constraints.

#### 3.2.1 Model 1

For ease of understanding we use the mathematical notation and modeling convention in the literature (Keskin & Çatay, 2016; Rastani et al., 2019; Schneider et al., 2014). We define  $V = \{1, \dots, n\}$  and  $F$  as the set of customers and recharging stations, respectively. Vertex 0 denotes the depot where each EV departs from and return to at the end of its tour. We create dummy copies of the depot in order to keep track of the battery SoC of each vehicle upon their return. Let  $AD$  denote the set of arrival depots and  $V_0 = V \cup \{0\}$ ,  $V_{AD} = V \cup AD$ ,  $V_{0,AD} = V \cup \{0\} \cup AD$ . Then, the problem can be represented on a complete directed graph  $G = (N, A)$  with the set of arcs  $A = \{(i, j) | i, j \in N, i \neq j\}$ , where set  $N = V_{0,AD} \cup F$  is the total set of nodes on the network.

The energy consumption depends on the distance traveled and the total weight of the EV, which includes the weight of the cargo. Each customer  $i \in V$  is associated with service time  $s_i$ , time window  $[e_i, l_i]$ , and demand  $q_i$ . All EVs have a battery capacity of  $Q$  and a cargo capacity of  $C$ . At each recharging station, one unit of energy is transferred in  $g$  time units.  $d_{ij}$  represents the direct distance from node  $i$  to  $j$ . Travel time from customer  $i$  to customer  $j$  is denoted by  $t_{ij}$  if the journey is direct and  $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$  is the extra travel time if it is via station  $s$ . Note that  $\hat{t}_{ijs}$  does not include the recharging time at station  $s$ .  $w$  represents the amount of additional energy required in order to move one unit of cargo. The total energy consumption of an EV traveling from customer  $i$  to customer  $j$  is calculated as  $(h + wu_i)d_{ij}$ , where  $u_i$  is the weight of the load on the vehicle upon departure from customer  $i$ .

The decision variables  $y_i, y_{ijs}$ , and  $Y_{ijs}$  keep track of battery SoC at arrival at customer/depot  $i$ , at arrival at station  $s$  on route  $(i, s, j)$ , and at departure from station  $s$  on route  $(i, s, j)$ , respectively.  $\tau_i$  denotes the time when the service starts at customer  $i$ . The binary decision variable  $x_{ij}$  takes a value of 1 if a vehicle travels from customer  $i$  to customer  $j$ ; and 0 otherwise. The binary decision variable  $z_{ijs}$  is equal to 1 if the journey from customer  $i$  to customer  $j$  is via station  $s$ .

The problem can be formulated as mixed-integer linear program as follows:

$$\text{Min } (y_0 - y_{AD}) + \sum_{i \in V_0} \sum_{j \in V_{AD}} \sum_{s \in F} (Y_{ijs} - y_{ijs}) \tag{5}$$

$$\text{subject to} \tag{6}$$

$$y_0 = Q \tag{6}$$

$$\sum_{\substack{j \in V_{AD} \\ j \neq i}} x_{ij} = 1 \quad \forall i \in V \tag{7}$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} = 1 \quad \forall j \in V_{AD} \tag{8}$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V_{AD} \\ i \neq j}} x_{ji} = 0 \quad \forall j \in V \quad (9)$$

$$\sum_{s \in F} z_{ijs} \leq x_{ij} \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (10)$$

$$\tau_i + (t_{ij} + r_i)x_{ij} + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs} + g(Y_{ijs} - y_{ijs})) - l_0(1 - x_{ij}) \leq \tau_j \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (11)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (12)$$

$$0 \leq y_j \leq y_i - (h + wu_i)d_{ij} + M(1 - x_{ij} + \sum_{s \in F} z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (13)$$

$$y_j \leq Y_{ijs} - (h + wu_i)d_{sj} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (14)$$

$$0 \leq y_{ijs} \leq y_i - (h + wu_i)d_{is} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (15)$$

$$y_{ijs} \leq Y_{ijs} \leq Qz_{ijs} \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (16)$$

$$y_j \leq Q \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} \quad \forall j \in V_{AD} \quad (17)$$

$$u_j \geq u_i + q_j x_{ij} - C(1 - x_{ij}) \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (18)$$

$$0 \leq u_i \leq C \quad \forall i \in V_{0,AD} \quad (19)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (20)$$

$$z_{ijs} \in \{0, 1\} \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (21)$$

The objective function (5) minimizes the total energy consumption. Since the EVs are assumed to recharge overnight, constraints (6) set their initial battery SoC to full. Constraints (7) and (8) impose the connectivity of customer visits whereas the flow conservation at each vertex is ensured by constraints (9). Constraints (10) make sure that customer  $j$  is served after customer  $i$  if the vehicle travels from  $i$  to  $j$  by recharging its battery en-route. Time feasibility of arcs emanating from the customers (the depot) is guaranteed by Constraints (11) whereas the time window restrictions are established by Constraints (12). Constraints (11) and (12) also prevent the formation of sub-tours. Constraints (13) keep track of the SoC when the vehicle travels from customer  $i$  to customer  $j$  without recharging en-route. Constraints (14) determine battery SoC at the arrival at customer  $j$  if the vehicle visits a recharging station after it has departed from customer  $i$ . Constraints (15) check the SoC at the arrival at a station if the vehicle recharges en-route. Here,  $M$  is a sufficiently large constant and can be  $M = Q + (h + w \sum_{i \in V} q_i) \times \max_{i \in V_0, j \in V_{AD}} \{d_{ij}\}$ . Constraints (16) set the upper bound for battery SoC when departing from a station. Constraints (17) allow positive battery SoC at the arrival at customer  $j$ , when customer  $j$  is visited. Constraints (18) keep track of the load of the vehicle throughout its tour. Constraints (19) ensure the non-negativity of the load on the vehicle and guarantee that the cargo capacity is not exceeded. Finally, constraints (20) and (21) define the binary decision variables.

### 3.2.2 Model 2

In Model 1, the objective function minimizes the total energy consumption calculated in terms of the changes in battery SoCs. Alternatively, the total energy consumption can be calculated by multiplying the lengths of the arcs traversed by the energy consumption rate per unit distance which vary by the weight of cargo carried onboard. Since the cargo load of an EV and the arcs traversed are associated with decision variables, the resulting objective function is nonlinear. The constraints, hence the feasible region, remain the same; however,  $AD = \{n + 1\}$  since copies of the depot are no longer necessary to keep track of the battery SoC of the EVs when they return to the depot. The mixed-integer nonlinear program can be formulated as follows:

$$\text{Min} \sum_{i \in V_0} \sum_{j \in V_{n+1}} [(h_{ij} + wu_i d_{ij})x_{ij} + \sum_{s \in F} (\hat{h}_{ijs} + wu_i \hat{d}_{ijs})z_{ijs}] \tag{22}$$

subject to (6)–(7), (9)–(21)

Since all constraints are linear, the feasible region is convex. A local minimum of a convex function on a convex feasible region is also a global minimum (Bertsekas et al., 1999). In order to check the convexity of the objective function, consider a small instance that involves only a depot and a customer. The objective function can be formulated as follows:

$$\text{Min}(h_{0,1} + wu_0 d_{0,1})x_{0,1} + (h_{1,0} + wu_1 d_{1,0})x_{1,0} \tag{23}$$

Since the distances are symmetric, we let  $d = d_{0,1} = d_{1,0}$  and  $h = h_{0,1} = h_{1,0}$  for simplicity. Hessian matrix comprises geometric information about the surface  $z = f(x, y)$  and is defined as follows:

$$H_f(x, y) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$$

where all of the second partial derivatives of function  $f$  exist at any point. Let  $H$  show the hessian matrix of the function in Eq. (23):

$$\begin{matrix} & x_{0,1} & x_{1,0} & u_0 & u_1 \\ x_{0,1} & \begin{bmatrix} 0 & 0 & wd & 0 \\ 0 & 0 & 0 & wd \\ wd & 0 & 0 & 0 \\ 0 & wd & 0 & 0 \end{bmatrix} & & & \\ x_{1,0} & & & & \\ u_0 & & & & \\ u_1 & & & & \end{matrix}$$

We calculate the eigenvalues of  $H$  in order to obtain the geometric information about the surface of our objective function. If all the eigenvalues are positive (negative), the function is convex (concave). If some of the eigenvalues are positive and some are negative, the function is neither convex nor concave, and it will have a saddle point. We calculate  $\det(H - I\lambda) = 0$ , where  $\lambda$  represent the eigenvalues, as follows:

$$\det \left( \begin{bmatrix} 0 & 0 & wd & 0 \\ 0 & 0 & 0 & wd \\ wd & 0 & 0 & 0 \\ 0 & wd & 0 & 0 \end{bmatrix} - \begin{bmatrix} -\lambda & 0 & 0 & 0 \\ 0 & -\lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & -\lambda \end{bmatrix} \right) = \det \left( \begin{bmatrix} -\lambda & 0 & wd & 0 \\ 0 & -\lambda & 0 & wd \\ wd & 0 & -\lambda & 0 \\ 0 & wd & 0 & -\lambda \end{bmatrix} \right) \\ = (\lambda^2 - (wd)^2)^2 = 0 \rightarrow \lambda = \pm wd$$

$w$  and  $d$  are positive scalars, thus  $\lambda$  values can be positive or negative. Hence, the objective function is neither convex nor concave. Consequently, the solution obtained by solving Model 2 provides a local minimum.

### 3.2.3 Analysis on the performance of the proposed models

To investigate the performances of Models 1 and 2 we solved the small-size EVRPTW dataset of Schneider et al. (2014). This dataset consists of 36 instances involving 5, 10, and 15 customers generated based on the VRPTW instances of Solomon (1987). The instances are classified according to the geographic distribution of the customers: clustered (c-type), random (r-type), and half clustered half random (rc-type). Furthermore, in type-1 problems (i.e., subsets r1, c1, rc1) the planning horizon is short, and the customers have narrow time windows compared to long planning horizon and wide time windows in type-2 problems (i.e., r2, c2, rc2). All instances were solved using Gurobi 9.0 with a 2-h time limit. The results are reported in Table 3. In this table, “#Veh”, “EC”, “ $t(sec)$ ”, and “Gap (%)” refer to the fleet size, energy consumption, run time (in seconds), and the reported optimality gap by Gurobi, respectively. “% $\Delta EC$ ” shows the percentage deviation of the energy consumption in Model 1 from that in Model 2. Note that the deviation is calculated for solutions with the same fleet size and a positive number indicates that Model 2 provides a better result. “Avg” stands for average.

The results show that the nonlinear Model 2 is capable of providing better solutions in shorter run time. It outperforms Model 1 in two instances in terms of #Veh and in two instances in terms of EC (shown in bold). Moreover, Model 2 required 84% less computational effort than Model 1 on the average. Although Model 2 does not guarantee a global minimum, the results reveal that it is more promising than Model 1 in terms of both the solution quality and run time.

The optimality gap for Model 2 in 34 out of 36 instances is 0 since Gurobi found the optimal solution; however, for Model 1 Gurobi stops when it reaches the time limit (7200 s) in 12 instances. We also increased the computational time for the two instances namely “r202c15-s6” and “rc204c15-s7” to 10 h using Model 2. The instance “r202c15-s6” proved that the provided solution in Table 3 is optimal in 2 h and 23 min. However, for the instance “rc204c15-s7”, it cannot decrease the gap that much and after 10 h it provides the same solution reported in Table 3 with the optimality gap of 14.53%. Encouraged by the performance of Model 2 on small-size problems, we selected a total of 18 instances involving 25- and 50- customers and performed additional tests to investigate the tractability of the problem. In the 25-customer instances Gurobi was able to solve only one out of 9 instances to optimality and the average optimality gap is 61.90%, whereas none of the 50-customer instances was solved optimally and the average optimality gap is 78.91%. These results show the complexity of the problem and highlight the need for a heuristic method to solve not only large-size but medium-size instances as well.

## 4 Solution methodology

Since the problem is intractable for large-size instances, we develop an LNS-based matheuristic approach to solve it. Introduced by Shaw (1998) LNS attempts to improve an incumbent solution using an iterative destroy-and-repair framework. In every iteration, some customers are removed from the solution using a destroy mechanism and reinserted into the routes using

**Table 3** Comparison of results obtained using Models 1 and 2

Instance	Model 1				Model 2				%ΔEC
	#Veh	EC	t (sec)	Gap (%)	#Veh	EC	t (sec)	Gap (%)	
r104c5-s3	2	141.54	< 1	0.00	2	141.54	< 1	0.00	0.00
r105c5-s3	2	159.23	< 1	0.00	2	159.23	< 1	0.00	0.00
r202c5-s3	1	144.12	< 1	0.00	1	144.12	< 1	0.00	0.00
r203c5-s4	1	181.32	< 1	0.00	1	181.32	< 1	0.00	0.00
c101c5-s3	2	266.02	< 1	0.00	2	266.02	< 1	0.00	0.00
c103c5-s2	1	186.83	< 1	0.00	1	186.83	< 1	0.00	0.00
c206c5-s4	1	250.63	< 1	0.00	1	250.63	< 1	0.00	0.00
c208c5-s3	1	168.91	< 1	0.00	1	168.91	< 1	0.00	0.00
rc105c5-s4	2	256.62	< 1	0.00	2	256.62	< 1	0.00	0.00
rc108c5-s4	2	264.00	< 1	0.00	2	264.00	< 1	0.00	0.00
rc204c5-s4	1	188.59	< 1	0.00	1	188.59	< 1	0.00	0.00
rc208c5-s3	1	170.82	< 1	0.00	1	170.82	< 1	0.00	0.00
Avg	1.42	198.22	0.25	0.00	1.42	198.22	0.04	0.00	0.00
r102c10-s4	3	336.00	4	0.00	3	336.00	< 1	0.00	0.00
r103c10-s3	2	220.50	213	0.00	2	220.50	< 1	0.00	0.00
r201c10-s4	1	262.40	4	0.00	1	262.40	< 1	0.00	0.00
r203c10-s5	1	227.26	3297	0.00	1	227.26	< 1	0.00	0.00
c101c10-s5	3	410.10	11	0.00	3	410.10	< 1	0.00	0.00
c104c10-s4	2	305.64	7200	94.17	2	305.64	1	0.00	0.00
c202c10-s5	1	319.04	< 1	0.00	1	319.04	< 1	0.00	0.00
c205c10-s3	2	233.74	18	0.00	2	233.74	< 1	0.00	0.00
rc102c10-s4	5	475.00	54	0.00	5	475.00	< 1	0.00	0.00
rc108c10-s4	3	364.62	46	0.00	3	364.62	< 1	0.00	0.00
rc201c10-s4	1	424.49	< 1	0.00	1	424.49	< 1	0.00	0.00
rc205c10-s4	2	334.54	103	0.00	2	334.54	< 1	0.00	0.00
Avg	2.17	326.11	913	7.85	2.17	326.11	0.92	0.00	0.00
r102c15-s8	5	430.83	7200	66.37	5	430.83	< 1	0.00	0.00
r105c15-s6	4	350.09	7200	47.08	4	350.09	< 1	0.00	0.00
r202c15-s6	2	365.06	7200	99.99	1	590.31	7200	7.47	–
r209c15-s5	1	347.31	7200	91.13	1	347.31	145	0.00	0.00
c103c15-s5	3	403.18	7200	89.78	3	<b>401.97</b>	27	0.00	0.30
c106c15-s3	3	352.29	2478	0.00	3	352.29	4	0.00	0.00
c202c15-s5	2	393.19	7200	99.32	2	393.19	2	0.00	0.00
c208c15-s4	2	310.12	7200	41.46	2	310.12	< 1	0.00	0.00
rc103c15-s5	4	415.79	7200	85.32	4	415.79	4	0.00	0.00

**Table 3** (continued)

Instance	Model 1				Model 2				%ΔEC
	#Veh	EC	t (sec)	Gap (%)	#Veh	EC	t (sec)	Gap (%)	
rc108c15-s5	4	430.69	7200	99.99	<b>3</b>	417.92	72	0.00	–
rc202c15-s5	2	403.03	7200	34.72	2	403.03	1	0.00	0.00
rc204c15-s7	1	402.41	7200	100.00	1	<b>402.15</b>	7200	16.95	0.06
Avg	2.75	383.67	6807	71.26	2.58	401.25	1221	2.03	0.04
Overall Avg	2.11	302.67	2573	26.37	2.06	308.53	407	0.67	0.01

a repair algorithm to generate a new, hopefully better solution. This procedure is repeated for a predetermined number of iterations. LNS and adaptive LNS (ALNS) have been successfully applied to many VRP variants including EVRPs and EVRPTWs (Goetze & Schneider, 2015; Hiermann et al., 2016; Keskin & Çatay, 2016, 2018; Keskin et al., 2019; Schiffer et al., 2018; Wen et al., 2016).

In this study, we generate the initial solution by adopting the insertion heuristic of Keskin and Çatay (2016) to the load-dependent case where the cost of inserting a customer into a route is calculated as  $(h + wu_i)d_{ik} + (h + wu_k)d_{kj} - (h + wu_i)d_{ij}$ . This insertion cost is calculated for all the customers that are not visited yet, and the minimum cost insertion is performed by ensuring that the constraints associated with the vehicle cargo capacity, battery SoC, customer time windows, and maximum tour duration are not violated. If an EV runs out of energy, a station may be inserted to make the tour energy feasible. We use first-feasible station insertion algorithm which will be described in Sect. 4.2. If no customer can be feasibly inserted into the route, a new route is initialized, and the procedure is repeated until all customers are served.

Our LNS employs several destroy and repair mechanisms adapted from the literature as well as new mechanisms that are designed to deal the more complex structure of the problem, including an optimal repair mechanism. In each iteration, a customer removal algorithm is applied to a feasible solution to remove a subset of customers from the routes. If any station is no longer needed in the partial solution, they are removed from the solution as well. Next, we apply a customer insertion algorithm that inserts all the removed customers to repair the solution with the aim of obtaining a new improved solution. Stations may be inserted to maintain the energy feasibility along the route. Solutions are accepted using simulated annealing rule. This procedure is repeated for a pre-determined number of iterations. Note that the set of stations that can be visited between any two customers is reduced by using the dominance rules presented in Bruglieri et al. (2016).

#### 4.1 Destroy operators

The current feasible solution is destroyed by removing  $\gamma$  customers. We use *worst-consumption*, *random worst-consumption*, *shaw*, *random worst-time*, *random*, *random route*, and *greedy route* removal procedures of Keskin and Çatay (2016) by modifying them for the load-dependent problem. At every iteration, one of these destroy operators is selected randomly.



- *Worst-consumption removal* algorithm selects the customers with high energy consumption imposed on the route by visiting that customer. We utilize the following selection criterion that calculates the energy consumption based on the distance travelled and cargo load:  $(h + wu_i)d_{ik} + (h + wu_k)d_{kj} - (h + wu_i)d_{ij}$ .
  - *Random worst-consumption removal* sorts the customers with respect to the associated energy consumptions, considers a subset of  $\sigma \times \gamma$  customers with the highest costs to select  $\gamma$  customers randomly, and removes them. Here,  $\sigma$  is a constant used to determine the size of the customer subset.
  - *Shaw removal* evaluates similar customers with respect to their energy consumption, earliest service time, being in the same route, and their demand. It randomly selects customer  $i$  and calculates the relatedness measure as  $R_{ij} = \phi_1 h_i d_{ij} + \phi_2 |e_i - e_j| + \phi_3 l_{ij} + \phi_4 |D_i - D_j|$  to determine other customers that are similar.  $\phi_1$ - $\phi_4$  are the shaw parameters,  $l_{ij} = -1$  if  $i$  and  $j$  are in the same route, 1 otherwise. Small  $R_{ij}$  means high similarity. Using the non-decreasing order of the customers according to their relatedness values with customer  $i$ , the first  $\gamma$  customers in the list are removed from the solution.
  - *Random worst-time removal* algorithm is a special case of shaw removal where  $\phi_1, \phi_3, \phi_4$  are set equal to 0. The customers are sorted in the non-decreasing order of their relatedness values. Then, from a subset of top  $\sigma \times \gamma$  customers in the list,  $\gamma$  customers are randomly selected and removed from the solution.
  - *Random removal* mechanism selects  $\gamma$  customers randomly and removes them from the solution.
  - *Random route removal* algorithm selects  $\omega$  routes randomly and removes them from the solution.
  - *Greedy route removal* mechanism sorts the routes in the non-decreasing order of the number of customers visited, and removes the first  $\omega$  routes that serve the least number of customers.
- Note that the route removal algorithms attempt to reduce the fleet size.

## 4.2 Repair operators

We adapt *random greedy*, *regret-2*, *random time-based*, *random greedy with noise function*, and *regret-2 with noise function* repair algorithms of Keskin and Çatay (2016) and Demir et al. (2012) for our load-dependent problem. In addition, we propose *exhaustive greedy*, *exhaustive time-based*, *exhaustive time-based with noise function*, and *random time-based with noise function* algorithms.

- *Random greedy insertion* selects a customer randomly and inserts it in the position that increases the energy consumption the least.
- *Random greedy with noise function insertion* is an extension of the *random greedy insertion* mechanism with a degree of freedom. We use the same noise function presented in Demir et al. (2012). The insertion cost using the freedom degree is calculated as  $NewCost = ActualCost + \bar{d}\mu\epsilon$ , where  $\bar{d}$  represents the maximum distance in the network,  $\mu$  is the noise parameter used for diversification, and  $\epsilon$  is a random number between  $[-1, 1]$ .
- *Exhaustive greedy insertion* considers all possible insertion positions for all customers that are not inserted yet and selects the customer-position matching which leads to least increase in energy consumption.
- *Exhaustive time-based insertion* calculates the difference between the route duration after and before inserting a customer as the insertion cost. For all customers, the insertion costs in all possible positions are calculated and the customer with the least insertion cost is inserted in the associated position.

- *Random time-based insertion* is similar to the *exhaustive time-based* algorithm. Instead of considering all unvisited customers, it first selects a customer randomly, and then inserts it in its best position with respect to time-based insertion cost.
- *Regret-2 insertion* aims at avoiding higher costs in the subsequent iterations. It calculates the difference between the cost of the best insertion and the second-best insertion for all customers, and insert the customer with the largest difference in the associated position.

Note that *regret-2 with noise function*, *exhaustive time-based with noise function*, *random time-based with noise function* are extensions of *regret-2*, *exhaustive time-based* and *random time-based* insertion mechanisms, respectively, using a similar noise function.

As we mentioned earlier, the unnecessary stations are removed from the partial solution obtained using the destroy operator. During the repair procedure, the insertion of a customer may not be feasible with respect to battery SoC. In that case, we first attempt to increase the recharge quantity if a station is visited prior to arriving at that customer. If the energy recharged at the station cannot be increased or no station is visited en-route, we apply a station insertion operator to make the insertion feasible. We modify the *greedy station insertion* and *best station insertion* operator of Keskin and Çatay (2016) and *multiple station insertion* operator of Rastani et al. (2019), and applied them for the load-dependent problem. Note that at most one station can be inserted between two consecutive customers in a route.

- *Greedy-station* insertion considers the first customer (or depot) where the EV arrives at with negative SoC and checks the insertion of the best station in the preceding arcs backward. The first station which makes the problem feasible is inserted.
- *Best-station insertion* algorithm determines the first customer (or depot) where the EV arrives at with negative SoC, evaluates the insertion costs on all the arcs from that customer backward along the route, and performs the insertion of the best station (which increases the consumption least) in its best position.
- *Multiple-station insertion* algorithm inserts multiple stations into a route when the insertion of a single station cannot make the route feasible. A station is inserted on the arc traversed immediately before arriving at the customer (or depot) with a negative SoC where the vehicle is recharged up to the maximum level allowed by the battery capacity and time windows restrictions of the succeeding customers. If the SoC is still negative at that customer or if the vehicle runs out of energy before reaching the inserted station, we attempt to insert another station prior to the last customer visited before traveling to the recently inserted station. This procedure is repeated until the route becomes energy feasible.

A station insertion algorithm is selected randomly when the repair procedure leads to infeasible route with respect to battery SoC. If one of the *first-feasible station insertion* or *best-station insertion* algorithms is selected but cannot create a feasible route, we resort to the *multiple-station insertion* algorithm. Furthermore, we remove all stations in the solution after every  $\beta$  iterations and apply the *best-station insertion* algorithm in an attempt to improve the solution.

### 4.3 Repair-opt insertion operator

We introduce a repair operator that attempts to insert the removed customers in the partial solution optimally along with the recharging stations, if needed. In this procedure, after the customers are removed from the solution, we feed the partial solution to a commercial solver and solve the mathematical programming model to insert them in their best positions in the solution. We refer to this operator as *repair-opt* procedure and implement it by employing

Model 2. We call *repair-opt* in the LNS framework after performing  $\eta$  iterations because: (i) it is expensive in terms of computational time, (ii) it is more promising if it is applied after the algorithm has converged to a good solution. Furthermore, in order to accelerate the solution time, we provide an upper bound on the number of vehicles. This upper bound is determined according to the feasible solution obtained using the LNS procedure and becomes tighter as the procedure progresses. The contribution of the proposed optimal repair mechanism on the performance of LNS is investigated in the experimental study.

## 5 Experimental study

We performed our computational tests using the dataset of Schneider et al. (2014) and Desaulniers et al. (2016) for the small- and large-size instances, respectively. The data characteristics are as described in Sect. 3.2.3. Schneider et al. (2014) generated 56 large-size instances based on the VRPTW instances of Solomon (1987). Desaulniers et al. (2016) modified them to maintain feasibility for the case where the vehicles are allowed to recharge at most once during their tours. We used this dataset in order to investigate the sensitivity of the results under different recharging scenarios. We focused on type-1 instances in the large-size dataset and omitted type-2 instances since wide time window constraints have minor influence on the recharging decisions. (Desaulniers et al., 2016; Keskin & Çatay, 2018).

In order to deal with a realistic vehicle cargo capacity and customer demands, we assumed an electric truck based on the specifications provided in Demir et al. (2012). Since the payload of this vehicle is 3650 kg, we converted the demand values to reasonable weights by multiplying each by “3650/original capacity” of the vehicle in order to observe the effect of cargo weight on energy consumption. We assumed a drive train efficiency of 0.9 as EVs are more efficient than internal combustion engine vehicles. Furthermore, since the EVs in the original data are assumed to consume one unit of energy per unit distance/time traveled, we used Eq. (4) to calculate the actual energy consumption of an empty vehicle (i.e., a truck with 6350 kg curb-weight) per unit distance and scaled it to  $h = 1$ . We used the same approach to determine the energy consumption  $w$  associated with unit load carried. For the sake of simplicity, we considered a flat terrain where road gradients are zero and the vehicle speed is constant and equals to 16.67 m/s.

The small-size instances were solved using Gurobi 9.0 with a 2-h time limit. LNS was coded in Python 3.7.1 and all runs were performed on an Intel Core(TM) i7-8700 processor with 3.20 GHz speed and 32 GB RAM. We performed five runs for each instance. Based on some preliminary experiments on a subset of instances, we observed that performing LNS for 1000 and 6000 iterations for the small- and large-size instances, respectively, provided good solutions in reasonable computational time. The values of the remaining parameters are given in Appendix A.

### 5.1 Results for small-size instances

We first consider the small-size instances to validate the performance of the proposed method and to investigate the influence of the cargo load on route plans. The results are reported in Table 4. In this table, the columns under “Gurobi” present the results provided by Gurobi and the columns under “LNS” give the results obtained by the proposed LNS algorithm. “Load Independent” column reports the results for the case that does not take into account the increased energy consumption associated with the cargo carried whereas column “Load

**Table 4** Results for small-size instances

Instance	Gurobi				LNS				
	Load independent		Load dependent		Load independent		Load dependent		
	#Veh	EC	t (sec)	%ΔEC	#Veh	EC	t (sec)	t (sec)	
r104c5-s3	2	136.69	<1	3.55	2	141.54	<1	141.54	1
r105c5-s3	2	156.08	<1	2.01	2	159.23	<1	159.23	1
r202c5-s3	1	128.88	<1	11.82	1	144.12	<1	144.12	2
r203c5-s4	1	179.06	<1	1.27	1	181.32	<1	181.32	2
c101c5-s3	2	257.75	<1	3.21	2	266.02	<1	266.02	1
c103c5-s2	1	175.37	<1	6.54	1	186.83	<1	186.83	1
c206c5-s4	1	242.56	<1	3.33	1	250.63	<1	250.63	2
c208c5-s3	1	164.34	<1	2.78	1	168.91	<1	168.91	2
re105c5-s4	2	233.77	<1	9.78	2	256.62	<1	256.62	1
re108c5-s4	2	253.93	<1	3.97	2	264.00	<1	264.00	1
re204c5-s4	1	185.16	<1	1.85	1	188.59	<1	188.59	3
re208c5-s3	1	167.98	<1	1.69	1	170.82	<1	170.82	2
Avg	1.42	190.13	<1	4.32	1.42	198.22	<1	198.22	2
r102c10-s4	3	249.19	<1	34.84	3	336.00	<1	336.00	3
r103c10-s3	2	206.30	1	6.88	2	220.50	<1	220.50	14
r201c10-s4	1	241.51	<1	8.65	1	262.40	<1	262.40	2
r203c10-s5	1	222.64	<1	2.08	1	227.26	<1	227.26	8
c101c10-s5	3	388.25	<1	5.63	3	410.10	<1	410.10	2
c104c10-s4	2	273.93	<1	11.58	2	305.64	1	305.64	85
c202c10-s5	1	304.06	<1	4.93	1	319.04	<1	319.04	1



Dependent” shows the results for the case that considers the load on the vehicle. The comparison of the result under these two columns exhibits the influence of the load on the fleet size and energy consumption. “#Veh”, “ $t$  (sec)”, and “ $EC$ ” refer to the fleet size, average run time (in seconds), and total energy consumption, respectively, as defined in Sect. 3.2.3, and “ $\% \Delta EC$ ” shows the percentage increase in the energy consumption for the load-dependent case as compared to the load-independent case.

The results show that Gurobi solved all the instances optimally except “rc204c15-s7” in the load-independent case, and “rc204c15-s7” and “r202c15-s6” in the load-dependent case. We observe that the existence of the load factor in the model required a larger fleet in one instance (“rc102c10-s4”) and resulted in an increase in the energy consumption in all instances, as expected. Furthermore, the energy consumption may be significantly higher when the cargo load is taken into account and the difference increases as the problem size increases. The actual energy consumption is 7.13% more on the average and can be up to 34.84% more due to the cargo load. When we compare the results found by our LNS approach to those of Gurobi, we see that LNS matches the optimal solutions (or the upper bounds in two instances) in all the instances except “r202c15-s6”. This constitutes an exceptional instance where a single EV serves all 15 customers in the optimal solution, which is difficult to identify heuristically.

## 5.2 Results for large-size instances

We solved the large-size instances for both load-independent and load-dependent cases using LNS. The results are provided in Table 5. We observe that the fleet size increases by two vehicles in one instance and by one vehicle in 15 instances (shown in bold) in the load-dependent case compared to the load-independent case. Furthermore, in the remaining 13 instances,  $EC$  increases by 13.11% on the average and can be as large as 31.63%. We believe that these significant differences in the energy consumptions and fleet sizes reveal the importance of considering cargo weight for more accurate route planning of EVs.

## 5.3 Sensitivity analysis

In this section, we first investigate the contribution of the repair-opt operator in LNS to the performance of the algorithm. Next, we analyze the sensitivity of the results to the single-recharge policy where an EV is allowed to recharge its battery at most once during its tour. This may be a practical case in urban logistics since the stations are scarce and may be unavailable, and the delivery companies do not want to waste the often-expensive vehicle and driver time with recharging.

### 5.3.1 Contribution of the insertion operators to algorithmic performance

In order to investigate the contribution of the insertion operators on the algorithmic performance we select 12 instances and re-solved them by removing one of the insertion operators at a time. We performed 5 runs and summarized the results in Table 6 in comparison with those reported in Table 5. “#Veh (%)” and “ $EC$  (%)” in Table 6 represent the average increase in fleet size and energy consumption when the corresponding insertion operator is omitted. These results show that the proposed LNS algorithm benefits from all of the insertion operators and the contribution of an operator to minimizing the fleet size and energy consumption can be

**Table 5** Results for large-size instances

Instance	Load independent			Load dependent			%ΔEC
	#Veh	EC	<i>t</i> (sec)	#Veh	EC	<i>t</i> (sec)	
r101	17	1679.69	782	<b>18</b>	1741.27	712	–
r102	16	1607.23	1761	<b>17</b>	1630.97	1135	–
r103	14	1277.51	2200	14	1404.57	1885	9.95
r104	12	1162.21	2776	<b>13</b>	1247.14	2545	–
r105	15	1348.54	1608	<b>16</b>	1482.75	1324	–
r106	14	1404.93	1885	<b>15</b>	1469.73	1752	–
r107	12	1243.62	2057	<b>13</b>	1332.27	1774	–
r108	12	1054.45	2836	12	1181.01	2560	12.00
r109	14	1261.83	1854	14	1415.5	1734	12.18
r110	13	1146.24	2656	13	1251.46	2129	9.18
r111	12	1193.67	2567	<b>13</b>	1262.93	2029	–
r112	12	1105.43	3616	12	1218.47	2763	10.23
Avg	13.58	1290.45	2217	14.17	1386.51	1862	10.71
c101	12	1043.38	682	12	1186.46	718	13.71
c102	11	1100.17	2252	<b>12</b>	1173.36	1969	–
c103	11	1246.22	3019	11	1395.65	3005	11.99
c104	11	1142.82	5640	11	1241.07	5024	8.60
c105	11	1037.26	1617	<b>12</b>	1161.57	1345	–
c106	11	1040.79	1896	<b>12</b>	1166.90	1919	–
c107	11	1021.93	2142	<b>12</b>	1179.50	1634	–
c108	12	1096.64	2570	12	1206.47	2466	10.02
c109	11	1001.64	3667	11	1318.41	2648	31.63
Avg	11.22	1081.21	2610	11.67	1225.49	2303	15.19
rc101	15	1652.33	1733	<b>17</b>	1921.03	1119	–
rc102	16	1607.69	1517	16	1815.88	1555	12.95
rc103	14	1501.91	1884	14	1722.56	1773	14.69
rc104	12	1380.05	2496	<b>13</b>	1515.47	2930	–
rc105	15	1539.88	2015	15	1734.78	1654	12.66
rc106	14	1484.98	1930	<b>15</b>	1650.57	1557	–
rc107	12	1264.11	2314	<b>13</b>	1467.85	2060	–
rc108	12	1398.05	2815	<b>13</b>	1533.09	2240	–
Avg	13.75	1478.63	2088	14.50	1670.15	1861	13.43
Overall Avg	12.85	1283.43	2305	13.44	1427.38	2009	13.11

as much as 3.69% (regret-2 insertion) and 6.82% (exhaustive greedy insertion), respectively, on average.

**Table 6** Average increases in fleet size and energy consumption upon removing an operator

Operator removed	Increase	
	#Veh (%)	EC (%)
Exhaustive greedy insertion	2.91	6.82
Random greedy insertion	3.09	0.32
Regret-2 insertion	3.69	4.24
Exhaustive time-based insertion	0.00	1.58
Random time-based insertion	0.60	2.02
Random greedy with noise function insertion	0.02	1.97
Regret-2 with noise function insertion	3.20	2.98
Exhaustive time-based with noise function insertion	0.44	2.42
Random time-based with noise function insertion	2.11	2.25
Repair-opt insertion	0.46	2.27

### 5.3.2 Contribution of repair-opt operator to algorithmic performance

Since the repair-opt operator is key component of the proposed method, we considered the LD-EVRPTW and re-solved all the instances by removing the repair-opt operator from the LNS to further investigate its role in the performance of the algorithm. We performed the LNS algorithm for 15,000 and 25,000 iterations to solve small- and large-size instances, respectively, as they showed a good compromise between the solution quality and computational effort. Note that the numbers of iterations were substantially smaller and set to 1000 and 6000 for small- and large-size problems, respectively, when LNS included the repair-opt.

We performed our tests on the 5-, 10-, 15-, and 100-customer instances. The results are summarized in Table 7. “#Better #Veh” and “#Better EC” reports the number of instances where LNS with repair-opt operator obtains a better solution with regard to the number of vehicles and energy consumption, respectively, and “#Inst” shows the number of instances solved. “% $\Delta EC$ ” represents the average percentage deviation in the energy consumption when LNS is performed without repair-opt operator as compared to with repair-opt. A negative value indicates the contribution of repair-opt to the solution quality.

The results show that the repair-opt operator allowed LNS to converge to better solutions in 8 small-size instances out of 36: in two instances it reduced the fleet size whereas in six instances it decreased the total energy consumption. In the large-size instances, the LNS equipped with repair-opt achieved smaller fleet size in two instances and provided savings in energy consumption in 17 instances. Note that the results with repair-opt were achieved with shorter computation times. The detailed results are reported in Appendix B. In light of these results, we conclude that the repair-opt procedure enhances the overall performance of the proposed LNS method.

### 5.3.3 Analysis of single-recharge strategy

To investigate the impact of the single-recharge (SR) strategy on the route plans and costs we solved all large-size problems by limiting the en-route recharges by one. Note that the results we provided earlier were obtained under the multiple recharge (MR) strategy where



**Table 7** Summary of results obtained without and with using repair-opt operator

Dataset	w/out Repair-opt			w/Repair-opt			t (sec)	%ΔEC	#Better #Veh	#Better EC
	#Inst	#Veh	EC	t (sec)	#Veh	EC				
5-Customer	12	1.42	198.33	11	1.42	198.33	2	0.00	0	0
10-Customer	12	2.33	315.17	22	2.17	326.08	11	-0.84	2	4
15-Customer	12	2.67	390.75	44	2.67	382.42	105	-1.78	0	2
100-Customer-R	12	14.25	1407.67	2326	14.17	1386.50	1862	-0.91	1	7
100-Customer-C	9	11.78	1254.33	2696	11.67	1225.44	2248	-2.91	1	8
100-Customer-RC	8	14.50	1664.50	2187	14.50	1670.25	1861	0.35	0	2

the EVs can recharge their batteries as many times as they need, but only once between two consecutive customers. Table 8 compares the results of the MR and SR strategies for the LD-EVRPTW. We observe that the SR strategy frequently leads to the utilization of additional vehicles in the fleet (in 18 out of 29 instances, as shown in bold). It also increases the energy consumption by up to 18.46%. These results show that the companies that adopt the SR

**Table 8** Comparison of results for multiple- and single-recharge strategies

Instance	MR			SR			%ΔEC
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	
r101	18	1741.27	712	<b>21</b>	1804.74	2845	–
r102	17	1630.97	1135	<b>18</b>	1625.74	2327	–
r103	14	1404.57	1885	<b>15</b>	1416.33	2347	–
r104	13	1247.14	2545	13	1286.17	2877	3.13
r105	16	1482.75	1324	<b>17</b>	1546.19	2740	–
r106	15	1469.73	1752	<b>16</b>	1503.25	2682	–
r107	13	1332.27	1774	<b>14</b>	1391.63	2437	–
r108	12	1181.01	2560	12	1234.29	2470	4.51
r109	14	1415.50	1734	<b>16</b>	1485.32	2441	–
r110	13	1251.46	2129	13	1291.98	2640	3.24
r111	13	1262.93	2029	<b>14</b>	1307.41	2819	–
r112	12	1218.47	2763	<b>13</b>	1244.18	2796	–
Avg	14.17	1386.51	1862	15.17	1428.10	2618	3.63
c101	12	1186.46	718	<b>13</b>	1305.60	1969	–
c102	12	1173.36	1969	12	1272.16	2576	8.42
c103	11	1395.65	3005	<b>12</b>	1263.82	2734	–
c104	11	1241.07	5024	11	1287.01	3238	3.70
c105	12	1161.57	1345	12	1375.97	2451	18.46
c106	12	1166.90	1919	12	1310.25	2576	12.28
c107	12	1179.50	1634	12	1232.97	2513	4.53
c108	12	1206.47	2466	12	1259.80	2474	4.42
c109	11	1318.41	2648	<b>12</b>	1211.23	2915	–
Avg	11.67	1225.49	2303	12.00	1279.87	2605	8.64
rc101	17	1921.03	1119	<b>18</b>	1979.08	2671	–
rc102	16	1815.88	1555	<b>17</b>	1859.46	2643	–
rc103	14	1722.56	1773	14	1753.40	2774	1.79
rc104	13	1515.47	2930	13	1521.31	2448	0.39
rc105	15	1734.78	1654	<b>16</b>	1862.04	2741	–
rc106	15	1650.57	1557	<b>16</b>	1768.59	2403	–
rc107	13	1467.85	2060	<b>14</b>	1539.99	2578	–
rc108	13	1533.09	2240	<b>14</b>	1569.16	2691	–
Avg	14.50	1670.15	1861	15.25	1731.63	2619	1.09
Overall Avg	13.44	1427.38	2009	14.14	1479.87	2614	4.45

strategy should re-evaluate this strategy by accounting accurately the trade-offs between driver/vehicle times and vehicle acquisition/fuel costs.

## 6 Conclusions and future research directions

In this paper, we addressed EVRPTW with partial recharges by taking into account the energy consumption associated with the cargo carried on the vehicle. We formulated two different mathematical models of the problem and compared their performances on small-size instances. For solving the large-size instances, we developed an LNS method by proposing new destroy and repair operators as well as a new repair mechanism based on an exact method.

Our numerical tests showed how the fleet size and/or energy consumption increase in comparison to the case where the load factor is neglected and revealed the importance of considering cargo weight of the vehicles for more accurate route planning. Specifically, we observed that the energy consumption can increase by up to 31% (in realistic large-size instances) because of the weight of the cargo carried on the vehicle. Furthermore, additional vehicles are often needed in order to provide service without disruption. Our experiments also revealed the contribution of the proposed repair-opt algorithm on the performance of the LNS algorithm.

Future research on this topic may consider the road gradient. A loaded vehicle going uphill will consume significantly more energy. On the other hand, when it travels downhill it can recharge its battery through energy recuperation from the wheels. However, the recuperation brings additional complexities to the problem.

**Acknowledgements** This research was supported by The Scientific and Technical Research Council of Turkey through Grant #118M412. This support is gratefully acknowledged. The authors also thank two anonymous referees for their valuable comments and suggestions.

## Appendix A: Parameter setting

The parameters used in the LNS algorithm are displayed in Table 9.

**Table 9** Parameter values

Par.	Description	Value
$\gamma$	Number of customers removed	Random between [20%, 55%]
$\Gamma$	Number of customers removed when repair-opt is selected as insertion operator	Random between [15%, 45%]
$\sigma$	Parameter used in random worst-consumption and random worst-time algorithm	1.5
$\omega$	Number of routes removed	Random between [10%, 40%]
$\phi_1$	First Shaw parameter	0.5
$\phi_2$	Second Shaw parameter	0.25

**Table 9** (continued)

Par.	Description	Value
$\phi_3$	Third Shaw parameter	0.15
$\phi_4$	Fourth Shaw parameter	0.25
$\mu$	Noise parameter	0.1
$\epsilon$	Random number for noise function	Random between $[-1, 1]$
$\alpha$	First-feasible station insertion selection probability	0.7
$\delta_1$	Worst-consumption removal selection probability	1/13
$\delta_2$	Random worst-consumption removal selection probability	4/13
$\delta_3$	Shaw removal selection probability	3/13
$\delta_4$	Random worst-time removal selection probability	1/13
$\delta_5$	Random removal selection probability	2/13
$\delta_6$	Random route removal selection probability	1/13
$\delta_7$	Greedy route removal selection probability	1/13
$\lambda_1$	Exhaustive greedy insertion selection probability	2/27
$\lambda_2$	Random greedy insertion selection probability	8/27
$\lambda_3$	Regret-2 insertion selection probability	4/27
$\lambda_4$	Exhaustive time-based insertion selection probability	2/27
$\lambda_5$	Random time-based insertion selection probability	2/27
$\lambda_6$	Random greedy with noise function insertion selection probability	2/27
$\lambda_7$	Regret-2 with noise function insertion selection probability	2/27
$\lambda_8$	Exhaustive time-based with noise function insertion selection probability	2/27
$\lambda_9$	Random time-based with noise function insertion selection probability	2/27
$\lambda_{10}$	Repair-opt insertion selection probability	1/27
$\beta$	Number of iterations to remove and reinsert stations	50
$\eta$	Number of iterations during which LNS is not allowed to call Repair-opt operator	800 for small-size instances 5800 for large-size instances

Note that values for the shaw and noise parameters were adopted from Demir et al. (2012) and the remaining parameters are determined by performing preliminary experiments. In these experiments,  $\delta$  and  $\lambda$  parameters were assigned weights varying from 1 to 4 and 1 to 8, respectively, and normalized to obtain probabilities whose sum is equal to 1.

## Appendix B. Detailed results analyzing the performance of repair-opt operator

The detailed results for the small- and large-size instances obtained using the LNS algorithm with and without employing the repair-opt operator for the load-dependent case are reported in Tables 10 and 11, respectively. In both tables, “#Veh”, “ $t$  (sec)”, and “EC” refer to the fleet size, average run time (in seconds), and total energy consumption, respectively, and “% $\Delta EC$ ” shows the percentage increase in the energy consumption when LNS does not employ the

**Table 10** Results for small-size instances obtained with and without using repair-opt

Instance	w/out Repair-opt			w/ Repair-opt			%ΔEC
	#Veh	EC	<i>t</i> (sec)	#Veh	EC	<i>t</i> (sec)	
r104c5-s3	2	142	11	2	142	1	0.00
r105c5-s3	2	159	8	2	159	1	0.00
r202c5-s3	1	144	16	1	144	2	0.00
r203c5-s4	1	181	9	1	181	2	0.00
c101c5-s3	2	266	9	2	266	1	0.00
c103c5-s2	1	187	10	1	187	1	0.00
c206c5-s4	1	251	10	1	251	2	0.00
c208c5-s3	1	169	9	1	169	2	0.00
rc105c5-s4	2	257	8	2	257	1	0.00
rc108c5-s4	2	264	10	2	264	1	0.00
rc204c5-s4	1	189	16	1	189	3	0.00
rc208c5-s3	1	171	14	1	171	2	0.00
r102c10-s4	4	272	22	<b>3</b>	336	3	–
r103c10-s3	2	220	29	2	220	14	0.00
r201c10-s4	1	270	14	1	<b>262</b>	2	– 2.82
r203c10-s5	1	227	4	1	227	8	0.00
c101c10-s5	3	428	20	3	<b>410</b>	2	– 4.21
c104c10-s4	2	308	48	2	<b>306</b>	85	– 0.77
c202c10-s5	1	321	9	1	<b>319</b>	1	– 0.62
c205c10-s3	2	234	26	2	234	3	0.00
rc102c10-s4	5	475	17	5	475	2	0.00
rc108c10-s4	3	365	23	3	365	4	0.00
rc201c10-s4	2	327	28	<b>1</b>	424	3	–
rc205c10-s4	2	335	25	2	335	4	0.00
r102c15-s8	5	431	28	5	431	5	– 0.04
r105c15-s6	4	350	32	4	350	4	0.00
r202c15-s6	2	365	30	2	365	272	0.00
r209c15-s5	1	360	25	1	<b>347</b>	36	– 3.52
c103c15-s5	3	402	73	3	402	419	– 0.01
c106c15-s3	3	352	52	3	352	12	0.00
c202c15-s5	2	393	44	2	393	33	0.00
c208c15-s4	2	310	47	2	310	15	0.00
rc103c15-s5	4	416	45	4	416	57	0.00
rc108c15-s5	3	418	40	3	418	20	0.00
rc202c15-s5	2	403	48	2	403	11	0.01
rc204c15-s7	1	489	58	1	<b>402</b>	371	– 17.76

**Table 11** Results for large-size instances obtained with and without using repair-opt

Instance	w/out Repair-opt			w/ Repair-opt			%ΔEC
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	
r101	19	1833	1357	<b>18</b>	1741	712	–
r102	17	1746	1430	17	<b>1631</b>	1135	– 6.57
r103	14	1375	1898	14	1405	1885	2.13
r104	13	1253	3017	13	<b>1247</b>	2545	– 0.49
r105	16	1481	1629	16	1483	1324	0.12
r106	15	1469	1928	15	1470	1752	0.06
r107	13	1357	2148	13	<b>1332</b>	1774	– 1.83
r108	12	1190	3596	12	<b>1181</b>	2560	– 0.73
r109	14	1457	1818	14	<b>1416</b>	1734	– 2.87
r110	13	1256	2967	13	<b>1251</b>	2129	– 0.36
r111	13	1271	2601	13	<b>1263</b>	2029	– 0.66
r112	12	1204	3521	12	1218	2763	1.19
c101	12	1198	1454	12	<b>1186</b>	718	– 0.93
c102	12	1210	2008	12	<b>1173</b>	1969	– 3.05
c103	11	1433	2809	11	<b>1396</b>	3005	– 2.61
c104	11	1456	5247	11	<b>1241</b>	4524	– 14.77
c105	12	1167	1934	12	<b>1162</b>	1345	– 0.50
c106	12	1171	2348	12	<b>1167</b>	1919	– 0.39
c107	12	1183	2490	12	<b>1180</b>	1634	– 0.32
c108	12	1215	3060	12	<b>1206</b>	2466	– 0.70
c109	12	1256	2913	<b>11</b>	1318	2648	–
rc101	17	1947	1485	17	<b>1921</b>	1119	– 1.34
rc102	16	1812	1627	16	1816	1555	0.23
rc103	14	1653	1975	14	1723	1773	4.23
rc104	13	1490	3282	13	1515	2930	1.71
rc105	15	1734	1845	15	1735	1654	0.02
rc106	15	1630	2008	15	1651	1557	1.25
rc107	13	1523	2486	13	<b>1468</b>	2060	– 3.65
rc108	13	1527	2786	13	1533	2240	0.37

repair-opt operator as compared to when it employs it. In Tables 10 and 11, the improvements in terms of energy consumption and number of vehicles are shown in bold.

## References

- Asamer, J., Graser, A., Heilmann, B., & Ruthmair, M. (2016). Sensitivity analysis for energy demand estimation of electric vehicles. *Transportation Research Part D: Transport and Environment*, 46, 182–199. <https://doi.org/10.1016/j.trd.2016.03.017>
- Bertsekas, D. P., Hager, W., & Mangasarian, O. (1999). *Nonlinear programming*. Athena Scientific Belmont.

- Breunig, U., Baldacci, R., Hartl, R. F., & Vidal, T. (2019). The electric two-echelon vehicle routing problem. *Computers and Operations Research*, 103, 198–210. <https://doi.org/10.1016/j.cor.2018.11.005>
- Bruglieri, M., Mancini, S., Pezzella, F., & Pisacane, O. (2016). A new mathematical programming model for the green vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 55, 89–92. <https://doi.org/10.1016/j.endm.2016.10.023>
- Bruglieri, M., Pezzella, F., Pisacane, O., & Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47, 221–228. <https://doi.org/10.1016/j.endm.2014.11.029>
- Çatay, B., & Keskin, M. (2017). The impact of quick charging stations on the route planning of electric vehicles. In *2017 IEEE Symposium on Computers and Communications (ISCC)* (pp. 152–157). IEEE. <https://doi.org/10.1109/ISCC.2017.8024521>
- Conrad, R. G., & Figliozzi, M. A. (2011). The recharging vehicle routing problem. In *Proceedings of the 2011 industrial engineering research conference*. IISE Norcross, GA.
- Corberán, Á., Erdoğan, G., Laporte, G., Plana, I., & Sanchis, J. M. (2018). The Chinese Postman problem with load-dependent costs. *Transportation Science*, 52(2), 370–385. <https://doi.org/10.1287/trsc.2017.0774>
- 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. <https://doi.org/10.1016/j.ejor.2012.06.044>
- Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6), 1388–1405. <https://doi.org/10.1287/opre.2016.1535>
- Edelstein, S. (2016). Netherlands joins Norway in plans to end new gas, diesel car sales by 2025, <[http://www.greencarreports.com/news/1103507\\_netherlands-joins-norway-in-plans-to-end-new-gas-diesel-car-sales-by-2025](http://www.greencarreports.com/news/1103507_netherlands-joins-norway-in-plans-to-end-new-gas-diesel-car-sales-by-2025)> (last accessed 08 Oct 2020).
- Erdelić, T., & Carić, T. (2019). A survey on the electric vehicle routing problem: Variants and solution approaches. *Journal of Advanced Transportation*. <https://doi.org/10.1155/2019/5075671>
- Felipe, Á., 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. <https://doi.org/10.1016/j.tre.2014.09.003>
- Froger, A., Mendoza, J. E., Jabali, O., Laporte, G. (2017). *A matheuristics for the electric vehicle routing problem with capacitated charging stations*. Technical Report, CIRRELT.
- Froger, A., Mendoza, J. E., Jabali, O., & Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers and Operations Research*, 104, 256–294. <https://doi.org/10.1016/j.cor.2018.12.013>
- Fukasawa, R., He, Q., & Song, Y. (2016). A branch-cut-and-price algorithm for the energy minimization vehicle routing problem. *Transportation Science*, 50(1), 23–34. <https://doi.org/10.1287/trsc.2015.0593>
- Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1), 81–99. <https://doi.org/10.1016/j.ejor.2015.01.049>
- Grandinetti, L., Guerriero, F., Pezzella, F., & Pisacane, O. (2016). A pick-up and delivery problem with time windows by electric vehicles. *International Journal of Productivity and Quality Management*, 18(2–3), 403–423. <https://doi.org/10.1504/IJPM.2016.076719>
- Guo, J., Jiang, Y., Yu, Y., & Liu, W. (2020). A novel energy consumption prediction model with combination of road information and driving style of BEVs. *Sustainable Energy Technologies and Assessments*, 42, 100826. <https://doi.org/10.1016/j.seta.2020.100826>
- Hiermann, G., Hartl, R. F., Puchinger, J., & Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1), 235–248. <https://doi.org/10.1016/j.ejor.2018.06.025>
- Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3), 995–1018. <https://doi.org/10.1016/j.ejor.2016.01.038>
- Hof, J., Schneider, M., & Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, 97, 102–112. <https://doi.org/10.1016/j.trb.2016.11.009>
- Jie, W., Yang, J., Zhang, M., & Huang, Y. (2019). The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research*, 272(3), 879–904. <https://doi.org/10.1016/j.ejor.2018.07.002>
- Kara, I., Kara, B. Y., & Yetis, M. K. (2007). Energy minimizing vehicle routing problem. In *International Conference on Combinatorial Optimization and Applications* (pp. 62–71). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-540-73556-4\\_9](https://doi.org/10.1007/978-3-540-73556-4_9)

- Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65, 111–127. <https://doi.org/10.1016/j.trc.2016.01.013>
- Keskin, M., & Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers and Operations Research*, 100, 172–188. <https://doi.org/10.1016/j.cor.2018.06.019>
- Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers and Operations Research*, 107, 77–94. <https://doi.org/10.1016/j.cor.2019.02.014>
- Khan, S. (2016). Germany pushes to ban petrol-fuelled cars within next 20 years. The Independent, <<http://www.independent.co.uk/news/world/europe/germany-petrol-car-ban-no-combustion-diesel-vehicles-2030-a7354281.html>> (last accessed 08 Oct 2020).
- Kopfer, H., & Vornhusen, B. (2019). Energy vehicle routing problem for differently sized and powered vehicles. *Journal of Business Economics*, 89(7), 793–821. <https://doi.org/10.1007/s11573-018-0910-z>
- Kullman, N., Goodson, J. and Mendoza, J.E. (2019). *Electric Vehicle Routing with Public Charging Stations*. Technical report hal-01928730v2, available at: <https://hal.archives-ouvertes.fr/hal-01928730v2/document>
- Li, C., Ding, T., Liu, X., & Huang, C. (2018). An electric vehicle routing optimization model with hybrid plug-in and wireless charging systems. *IEEE Access*, 6, 27569–27578. <https://doi.org/10.1109/ACCESS.2018.2832187>
- Liao, C. S., Lu, S. H., & Shen, Z. J. M. (2016). The electric vehicle touring problem. *Transportation Research Part B: Methodological*, 86, 163–180. <https://doi.org/10.1016/j.trb.2016.02.002>
- Liu, K., Wang, J., Yamamoto, T., & Morikawa, T. (2018). Exploring the interactive effects of ambient temperature and vehicle auxiliary loads on electric vehicle energy consumption. *Applied Energy*, 227, 324–331. <https://doi.org/10.1016/j.apenergy.2017.08.074>
- Liu, R., & Jiang, Z. (2019). A constraint relaxation-based algorithm for the load-dependent vehicle routing problem with time windows. *Flexible Services and Manufacturing Journal*, 31(2), 331–353. <https://doi.org/10.1007/s10696-018-9323-0>
- Li-ying, W., & Yuan-bin, S. (2015). Multiple charging station location-routing problem with time window of electric vehicle. *Journal of Engineering Science and Technology Review*, 8(5), 190–201.
- Madankumar, S., & Rajendran, C. (2018). Mathematical models for green vehicle routing problems with pickup and delivery: A case of semiconductor supply chain. *Computers and Operations Research*, 89, 183–192. <https://doi.org/10.1016/j.cor.2016.03.013>
- Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part E: Logistics and Transportation Review*, 118, 392–420. <https://doi.org/10.1016/j.tre.2018.08.005>
- Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103, 87–110. <https://doi.org/10.1016/j.trb.2017.02.004>
- Morlock, F., Rolle, B., Bauer, M., & Sawodny, O. (2019). Forecasts of electric vehicle energy consumption based on characteristic speed profiles and real-time traffic data. *IEEE Transactions on Vehicular Technology*, 69(2), 1404–1418. <https://doi.org/10.1109/TVT.2019.2957536>
- Neubauer, J., & Wood, E. (2014). Thru-life impacts of driver aggression, climate, cabin thermal management, and battery thermal management on battery electric vehicle utility. *Journal of Power Sources*, 259, 262–275. <https://doi.org/10.1016/j.jpowsour.2014.02.083>
- Paz, J., Granada-Echeverri, M., & Escobar, J. (2018). The multi-depot electric vehicle location routing problem with time windows. *International Journal of Industrial Engineering Computations*, 9(1), 123–136. <https://doi.org/10.5267/j.ijiec.2017.4.001>
- Pelletier, S., Jabali, O., & Laporte, G. (2016). 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, 50(1), 3–22. <https://doi.org/10.1287/trsc.2015.0646>
- Pelletier, S., Jabali, O., Laporte, G., & Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transportation Research Part B: Methodological*, 103, 158–187. <https://doi.org/10.1016/j.trb.2017.01.020>
- Rastani, S., Yüksel, T., & Çatay, B. (2019). Effects of ambient temperature on the route planning of electric freight vehicles. *Transportation Research Part D: Transport and Environment*, 74, 124–141. <https://doi.org/10.1016/j.trd.2019.07.025>
- Sassi, O., Cherif, W. R., & Oulamara, A. (2015). Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. *International Journal of Mathematical and Computational Sciences*, 9(3), 171–181. <https://doi.org/10.5281/zenodo.1099756>



- Schiffer, M., Schneider, M., & Laporte, G. (2018). Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. *European Journal of Operational Research*, 265(2), 517–532. <https://doi.org/10.1016/j.ejor.2017.07.067>
- Schiffer, M., & Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260(3), 995–1013. <https://doi.org/10.1016/j.ejor.2017.01.011>
- Schneider, M., Stenger, A., & Goetze, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4), 500–520. <https://doi.org/10.1287/trsc.2013.0490>
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming* (pp. 417–431). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/3-540-49481-2\\_30](https://doi.org/10.1007/3-540-49481-2_30)
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254–265. <https://doi.org/10.1287/opre.35.2.254>
- Taş, D. (2020). Electric vehicle routing with flexible time windows: a column generation solution approach. *Transportation Letters*. <https://doi.org/10.1080/19427867.2020.1711581>
- Wang, Y. W., Lin, C. C., & Lee, T. J. (2018). Electric vehicle tour planning. *Transportation Research Part D: Transport and Environment*, 63, 121–136. <https://doi.org/10.1016/j.trd.2018.04.016>
- Wen, M., Linde, E., Ropke, S., Mirchandani, P., & Larsen, A. (2016). An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers and Operations Research*, 76, 73–83. <https://doi.org/10.1016/j.cor.2016.06.013>
- Wu, X., Freese, D., Cabrera, A., & Kitch, W. A. (2015). Electric vehicles' energy consumption measurement and estimation. *Transportation Research Part D: Transport and Environment*, 34, 52–67. <https://doi.org/10.1016/j.trd.2014.10.007>
- Wyman, O. (2018). Governmental incentives to encourage electric-vehicle sales and use. < <https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/march/Governmental-Incentives-To-Encourage-%20Electric-Vehicle-Sales-And-Use-exhibit.pdf>> (last accessed 08 Oct 2020).
- Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and Operations Research*, 39(7), 1419–1431. <https://doi.org/10.1016/j.cor.2011.08.013>
- Yang, J., & Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. *Computers and Operations Research*, 55, 217–232. <https://doi.org/10.1016/j.cor.2014.07.003>
- Yuksel, T., & Michalek, J. J. (2015). Effects of regional temperature on electric vehicle efficiency, range, and emissions in the United States. *Environmental Science and Technology*, 49(6), 3974–3980. <https://doi.org/10.1021/es505621s>
- Yuksel, T., Tamayao, M. A. M., Hendrickson, C., Azevedo, I. M., & Michalek, J. J. (2016). Effect of regional grid mix, driving patterns and climate on the comparative carbon footprint of gasoline and plug-in electric vehicles in the United States. *Environmental Research Letters*, 11(4), 044007. <https://doi.org/10.1088/1748-9326/11/4/044007>
- Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2015). The load-dependent vehicle routing problem and its pick-up and delivery extension. *Transportation Research Part b: Methodological*, 71, 158–181. <https://doi.org/10.1016/j.trb.2014.11.004>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.