Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows

Merve Keskin and Bülent Çatay[1]

Sabanci University, Faculty of Engineering and Natural Sciences,
34956, Tuzla, Istanbul, Turkey
{catay, mervekeskin}@sabanciuniv.edu

Abstract

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) is an extension to the well-known Vehicle Routing Problem with Time Windows (VRPTW) where the fleet consists of electric vehicles (EVs). Since EVs have limited driving range due to their battery capacities they may need to visit recharging stations while servicing the customers along their route. The recharging may take place at any battery level and after the recharging the battery is assumed to be full. In this paper, we relax the full recharge restriction and allow partial recharging (EVRPTW-PR) which is more practical in the real world due to shorter recharging duration. We formulate this problem as 0-1 mixed integer linear program and develop an Adaptive Large Neighborhood Search (ALNS) algorithm to solve it efficiently. We apply several removal and insertion mechanisms by selecting them dynamically and adaptively based on their past performances, including new mechanisms specifically designed for EVRPTW and EVRPTW-PR. We test the performance of ALNS by using benchmark instances from the recent literature. The computational results show that the proposed method is effective in finding high quality solutions and the partial recharging option may significantly improve the routing decisions.

Keywords: Electric vehicle; vehicle routing problem with time windows; adaptive large neighborhood search; metaheuristics; partial recharge.

[1] Corresponding author. Tel: +90 (216) 483-9531, Fax: +90 (216) 483-9550.

1. Introduction

Transportation systems account for about 20-25% of global energy consumption and CO_2 emissions. Road transport is a major contributor with 75% share (White Paper on Transport, 2011). 95% of the world's transportation energy comes from fossil fuels, mainly gasoline and diesel. In the US, about 27% of total greenhouse gas (GHG) emissions in 2013 are transport related (www.epa.gov). 74% of the domestic freight in 2012 is moved by trucks and the freight volume is expected to grow by 39% in 2040 (Bureau of Transportation Statistics, 2014). Transport accounts for 63% of fuel consumption and 29% of all CO_2 emissions in the EU. Freight transport activity is predicted to grow by around 80% in 2050 compared to 2005 (ec.europa.eu).

Transportation will continue to be a major and still growing source of GHGs. Hence, governments are considering new environmental measures and targets for reducing emissions and fuel resource consumptions. The US Administration aims at cutting the overall GHG emissions 17% below 2005 levels by 2020 and has recently established the toughest fuel economy standards for internal combustion engine (ICE) vehicles in the US history (www.state.gov). The EU targets 80–95% reduction of GHGs below 1990 levels by 2050, where a reduction of at least 60% is expected from the transport sector. The European Commission aims at reducing the transport-related GHG emissions to around 20% below their 2008 level by 2030. The use of conventionally fuelled cars will be reduced by 50% in urban transport by 2030 and phased out by 2050. City logistics in major European urban centers will be CO_2-free by 2030 (White Paper on Transport, 2011).

The targets set by governments and the new regulations imposed encourage the usage of alternative fuel vehicles (AFV) such as solar, electric, biodiesel, LNG, CNG vehicles. Many municipalities, government agencies, non-profit organizations, and private companies are converting their fleets to include AFVs, either to reduce their environmental impact voluntarily or to meet new environmental regulations (Erdoğan and Miller-Hooks, 2012). Consequently, the advancements in the electric vehicle (EV) technology have gained momentum in parallel with the growing environmental concerns in societies.

EVs move with electric propulsion and can provide emission-free urban transportation. They can be classified as battery electric vehicles (BEV), hybrid electric vehicles (HEV), and fuel-cell electric vehicles (FCEV) such as electric trains, airplanes, boats, motorcycles, scooters, and spacecrafts (Chan, 2002). In this paper, we refer to EV as a commercial road BEV such as a lorry or van. A fleet of EVs can be used in a variety of transport needs such as public transportation, home deliveries from grocery stores, postal deliveries and courier services, distribution operations in different sectors.

Although EVs enable low-emission logistics services, operating an EV fleet has several drawbacks such as: (i) low energy density of batteries compared to the fuel of combustion engined vehicles; (ii) limited number of public charging stations; and (iii) long recharging times (Touati-Moungla and Jost, 2011). Battery swap may remedy the last; however, swapping raises additional issues in battery design and compatibility, battery degradation, ownership, and swap station infrastructure. Under these limitations, routing an EV fleet arises as a challenging combinatorial optimization problem in the Vehicle Routing Problem (VRP) literature.

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) was introduced by Schneider et al. (2014) as an extension to the Green Vehicle Routing Problem (GVRP) of Erdoğan and Miller-Hooks (2012). GVRP concerns "green" vehicles which run with biodiesel, liquid natural gas, or CNG, and have a limited driving range. Hence, the vehicles may need refueling along their route. Refueling is fast; however, the stations for these fuels are scarce. EVRPTW is a variant of the classical VRPTW where the fleet consists of EVs that may need to visit stations to have their batteries recharged in order to continue their route, as in GVRP. On the other hand, the recharging operation may take a significant amount of time, especially when compared to relatively short refueling times of liquid fuels. Recharging may take place at any battery level and the recharge time is proportional to the amount charged. After the recharge the battery is assumed to be full. The number of stations is usually small and the stations are dispersed in distant locations, which increases the difficulty of the problem. In this paper, we relax the full recharge restriction and allow partial recharging which is more practical in the real world due to shorter recharging duration. When the vehicle visits a station near the end of its route, full recharge may not be needed for the vehicle to return to the depot. A similar situation may exist between two consecutive recharges. Saving from recharging time may allow the vehicle to catch the time window of an otherwise unvisited customer, thus, may improve the solution.

In the partial recharge (PR) scheme, the recharge quantity is associated with a continuous decision variable. We refer to this problem as EVRPTW and Partial Recharges (EVRPTW-PR) and formulate it as 0-1 mixed integer linear program. Note that determining the recharge quantities brings significant difficulties to the problem. Since the problem is intractable for large instances, we propose an Adaptive Large Neighborhood Search (ALNS) approach to solve it efficiently. ALNS is based on the destroy-and-repair framework where at each iteration the existing feasible solution is destroyed by removing some customers and recharging stations from their routes and then repaired by inserting the removed customers to the solution along with stations when recharging is necessary. Several removal and insertion algorithms are applied by selecting them dynamically and adaptively based on their past performances**.** The new solution is accepted according to the Simulated Annealing criterion. Our approach combines the removal and insertion mechanisms presented in Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007) and Demir et al. (2012) with some new mechanisms designed specifically for EVRPTW and EVRPTW-PR. Our computational tests show that the proposed ALNS is effective in finding good quality solutions and improves some of the best-known solutions in the literature. Furthermore, our results reveal that the QF scheme may substantially improve the routing decisions.

The contributions of this study can be summarized as follows:

- We extend EVRPTW to a PR scheme, which is more general and practical, and present the mathematical programming formulation of the problem.
- We propose an effective ALNS method to solve the EVRPTW and EVRPTW-PR. The proposed method introduces new removal and insertion mechanisms to tackle the more complex problem structure of VRPs where the fleet consists of EVs.
- We validate the performance of the proposed method using the EVRPTW instances of Schneider et al. (2014) and improve the best-known solutions of 4 problems.
- We show that the PR scheme improves the solutions obtained with the full charging (FC) scheme substantially.

The remainder of the paper is organized as follows: Section 2 reviews the related studies in the literature. Section 3 describes the problem and formulates the mathematical model. The proposed ALNS method is presented in Section 4. Section 5 provides the computational study and discusses the results. Finally, concluding remarks and future research directions are given in Section 6.

2. Related literature

There are relatively few studies on route optimization of AFVs. Artmeirer et al. (2010) studied this problem within a graph-theoretic context and proposes extensions to general shortest path algorithms that address the problem of energy-optimal routing. They formalize energy-efficient routing in the presence of rechargeable batteries as a special case of the constrained shortest path problem and present an adaption of a general shortest path algorithm that respects the given constraints. Wang and Shen (2007) developed a model that minimizes the number of tours and total deadhead time hierarchically. The driving range of the vehicle is limited but the charging durations, time windows and vehicle capacities are not considered. A multiple ant colony algorithm was proposed to solve the problem.

Conrad and Figliozzi (2011) introduced the Recharging VRP (RVRP), a new variant of the VRP where the EVs are allowed to recharge at selected customer locations. The model has dual objectives: the primary objective minimizes the number of routes or vehicles whereas the secondary objective minimizes the total costs associated with the travel distance, service time and vehicle recharging. The latter is a penalty cost incurred at each recharge. The EV is charged while servicing the customer and the charging time is constant. The battery level departing from a customer depends on the choice of full charge or partial charging. In the partial charge case the battery is charged to a specified level such as 80% of battery capacity. Conrad and Figliozzi (2011) used an iterative construction and improvement procedure to solve this problem but did not provide its details.

Wang and Cheu (2012) investigated the operations of an electric taxi fleet. Their model minimizes total distance travelled under the recharging constraints and maximum route time. The battery is consumed at a given rate per distance and can be replenished at the recharging stations. Charging times are constant and after charging the battery becomes full. They construct an initial solution using one of the nearest-neighbor, sweep and earliest time window insertion heuristics and improve it using Tabu Search (TS). They also suggested three different recharging plans which provide different driving ranges and compared the results against the full charging scheme.

Omidvar and R. Tavakkoli-Moghaddam (2012) tackled an AFV routing problem with time-windows and proposed a mathematical model that minimizes total costs related to the vehicles, distance travelled, travel time and emissions. The refueling times are assumed constant. They used the Simulated Annealing (SA) and Genetic Algorithm (GA) approaches and compared their performances. Worley et al. (2012) addressed the problem of locating recharging stations and designing EV routes simultaneously. The objective is to minimize the sum of the travel costs, recharging costs, and costs of locating recharging stations. A solution method was not proposed and left as future work.

Erdoğan and Miller-Hooks (2012) considered the routing of AFVs within the context GVRP and formulated the mathematical model. The model aims at minimizing the total distance travelled where the length of the routes is restricted. Fuel is consumed with a given rate per traveled distance and can be replenished at the alternative fuel stations. Refueling times are assumed to be fixed and after refueling the tank becomes full. The model does not involve time windows and vehicle capacity constraints. Erdoğan and Miller-Hooks (2012) proposed two heuristics to solve the GVRP. The first is a Modified Clarke and Wright Savings (MCWS) algorithm which creates routes by establishing feasibility through the insertion of AFSs, merging feasible routes according to savings values, and removing redundant AFSs. The second is a Density-Based Clustering Algorithm (DBCA) based on the cluster-first and route-second approach. DBCA forms clusters of customers such that every vertex within a given radius contains at least a predefined number of neighbors. Subsequently, the MCWS algorithm is applied to the identified clusters. To test the performance of these two heuristics, they designed two sets of problem instances. The first consists of 40 small-sized instances with 20 customers while the second involves 12 instances with up to 500 customers.

Recently, Felipe et al. (2014) extended GVRP for EVs by allowing partial recharges using multiple technologies, i.e. using different power options. As in GVRP, the problem does not involve time windows but EVs have capacity and total route duration limits. The authors formulated the mathematical programming model and presented constructive and deterministic local search algorithms as well as a metaheuristic extension based on an SA framework. The computational tests on both randomly generated and GVRP data showed that using partial recharge strategies and providing multiple recharge technologies could achieve cost and energy savings and ensure feasibility in some instances.

Schneider et al. (2014) developed a hybrid metaheuristic that combines the Variable Neighborhood Search (VNS) algorithm with TS for solving EVRPTW. They tested the performance of the proposed method on benchmark instances of GVRP and Multi-Depot VRP with Inter-Depot Routes. They also generated new instances for EVRPTW by modifying Solomon (1987) data and reported their results. Desaulniers et al. (2014) tackled the same problem by considering four recharging strategies (single-full recharge, single-partial recharge, multiple-full recharge, and multiple-partial recharge) and attempted to solve them optimality using branch-price-and-cut algorithms. Goeke and Schneider (2015) extended EVRPTW to the routing of a mixed fleet of EVs and ICE vehicles. Their objective function consisted of an energy consumption function of speed, gradient, and cargo load distribution, and they proposed an

ALNS approach to solve it. Hiermann et al. (2015) addressed the Fleet Size and Mix Vehicle Routing Problem with Time Windows where the fleet consists of EVs. They also implemented an ALNS algorithm equipped with local search and labeling procedures.

ALNS was introduced by Ropke and Pisinger (2006a) as an extension of the Large Neighborhood Search (LNS) framework put forward by Shaw (1998). Ropke and Pisinger (2006b) developed a unified ALNS heuristic for a large class of VRP with Backhauls. Pisinger and Ropke (2007) improved this heuristic with additional algorithms and showed that the proposed framework gives competitive results in different VRP variants. Different implementations of ALNS for various VRP variants include cumulative capacitated VRP (Ribeiro and Laporte, 2012), pollution-routing problem (Demir et al., 2012), two-echelon VRP (Hemmelmayr et al., 2012), pickup and delivery problems with transshipment (Qu and Bard, 2012) and with vehicle transfers (Masson et al., 2013), VRP with multiple routes (Azi et al., 2014), periodic inventory routing problem (Aksen et al., 2014), and production routing problem (Adulyasak et al., 2014).

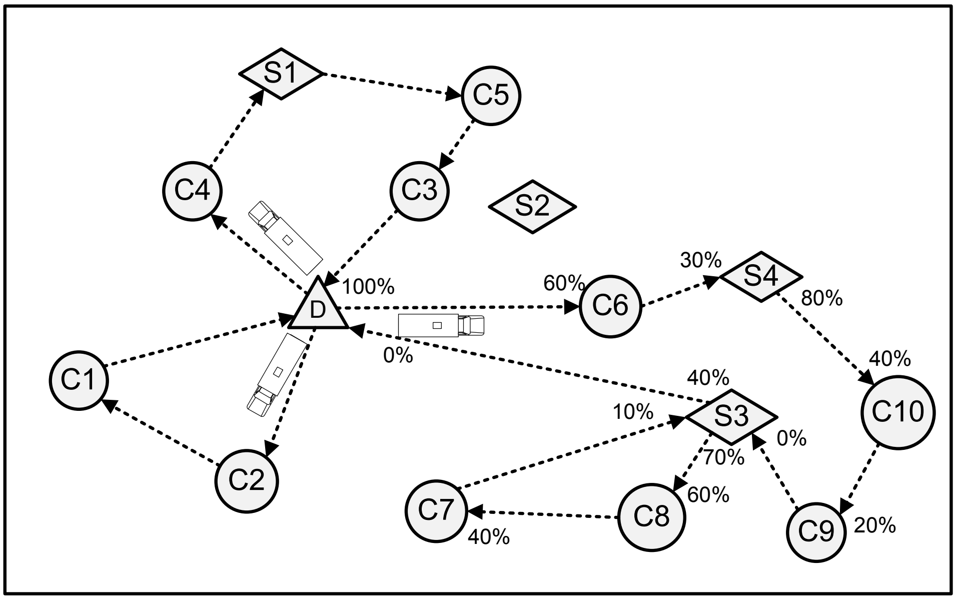


Figure 1. An illustrative example

3. Problem Description

EVRPTW-PR concerns a set of customers with known demands, delivery time window, and service durations. The deliveries are performed by a homogeneous fleet of EVs with fixed loading capacities and limited cruising ranges. While the vehicle is traveling, the battery charge level decreases proportionally with the distance traversed and the vehicle may need to visit a recharging station in order to continue its route. The battery is re charged at any quantity and the duration of the recharge depends on the initial state of battery charge. Unlike EVRPTW where the vehicle departs from the depot/station with full battery and arrives at the depot/station with any state of charge, in EVRPTW-PR the vehicle departs from the depot/station at any state of charge and arrives at the depot/station with an empty battery.

Fig. 1 illustrates an example involving 10 customers (C1-C10), 4 stations (S1-S4), and the depot (D) which can also be used for recharging. The percentage values along the route of EV3 show the battery state of charge when the vehicle arrives at a customer or a station and when it departs from the station after having its battery recharged. EV1 services C1 and C2, returns to the depot with its initial charge. EV2 visits S1 after servicing C4 and has its battery recharged before visiting C5 and C3. On the other hand, EV3 is recharged once in S4 and twice in S3. Note that a station can be visited multiple times by the same (see S3) or different vehicles and each station is not necessarily visited (see S2). In what follows, we provide the mathematical model for EVRPTW-PR following the formulation of Schneider et al. (2014).

Let $V = \{1, \dots, N\}$ denote the set of customers and F denote the set of recharging stations. Since a recharging station may be visited more than once depending on the route structure, we create $F^{'}$which is the set of dummy vertices generated to permit several visits to each vertex in the set F. Vertices 0 and $N+1$ denote the depot and every route starts at 0 and ends at $N+1$. Let $V^{'}$ be a set of vertices with $V^{'} = V \cup F^{'}$. In order to indicate that a set contains the respective instance of the depot, the set is subscripted with 0 or $N+1$. Hence $V_0^{'} = V^{'} \cup \{0\}$ and $V_{N+1}^{'} = V^{'} \cup \{N+1\}$. Now we can define the problem on a complete directed graph $G = (V_{0,N+1}^{'}, A)$ with the set of arcs $A = \{(i,j)|\ i,j \in V_{0,N+1}^{'},\ i \neq j\}$. Each arc is associated with a distance d_{ij} and travel time t_{ij}. The battery charge is consumed at a rate of r and every traveled arc consumes $r \cdot d_{ij}$ of the remaining battery. q is the recharge quantity in the quick charge option and is constant. Each vertex $i \in V^{'}$ has positive demand D_i, service time s_i and time window $[e_i, l_i]$. All EVs have a load capacity of C and battery capacity of Q. At a recharging station, the battery is charged at a recharging rate of g. The decision variables, τ_i, u_i and y_i keep track of the service starting time, remaining cargo level and remaining charge level at customer $i \in V_{0,N+1}^{'}$, respectively, and Y_i is the battery state of charge on departure from station $\forall i \in F^{'}$. The binary decision variable x_{ij} takes value 1 if arc (i,j) is traversed and 0 otherwise.

$$\min \quad \sum_{i\in V_0',j\in V_{n+1}',i\neq j} d_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{j\in V_{n+1}',i\neq j} x_{ij} = 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{j\in V_{n+1}',i\neq j} x_{ij} \leq 1 \qquad \forall i \in F' \tag{3}$$

$$\sum_{i\in V_0',i\neq j} x_{ij} = \sum_{i\in V_{n+1}',i\neq j} x_{ji} \qquad \forall j \in V' \tag{4}$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - l_0(1 - x_{ij}) \leq \tau_j \qquad \forall i \in V_0, \forall j \in V_{n+1}',\ i \neq j \tag{5}$$

$$\tau_i + t_{ij}x_{ij} + gq_i - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j \qquad \forall i \in F', \forall j \in V_{n+1}',\ i \neq j \tag{6}$$

$$e_j \leq \tau_j \leq l_j \qquad \forall j \in V_{0,n+1}' \tag{7}$$

$$0 \leq u_j \leq u_i - D_i x_{ij} + C(1 - x_{ij}) \qquad \forall i \in V_0', \forall j \in V_{n+1}',\ i \neq j \tag{8}$$

$$0 \leq u_0 \leq C \tag{9}$$

$$0 \leq y_j \leq y_i - (r \cdot d_{ij})x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in V, \forall j \in V_{n+1}',\ i \neq j \tag{10}$$

$$0 \leq y_j \leq Y_i - (r \cdot d_{ij})x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in F_0', \forall j \in V_{n+1}',\ i \neq j \tag{11}$$

$$Y_i = y_i + q_i \qquad \forall i \in F_0' \tag{12}$$

$$Y_i \leq Q \qquad \forall i \in F_0' \tag{13}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V_0', \forall j \in V_{n+1}',\ i \neq j \tag{14}$$

$$q_i \geq 0 \qquad \forall i \in F_0' \tag{15}$$

The objective function (1) minimizes total distance traveled. Constraints (2) and (3) handle the connectivity of customers and visits to recharging stations, respectively. The flow conservations constraints (4) enforce that the number of outgoing arcs equals to the number of incoming arcs at each vertex. Constraints (5) and (6) ensure the time feasibility of arcs leaving the customers (and the depot), and the stations, respectively. Constraints (7) enforce the time windows of the customers and the depot. In addition, constraints (5)-(7) eliminate the sub-tours. Constraints (8) and (9) guarantee that demand of all customers are satisfied. Constraints (10) and (11) keep track of the battery state of charge and make sure that it is never negative. Constraints (12) determine the battery state of charge after the recharge at a station. Constraints (13) make sure that the battery state of charge does not exceed its capacity. Finally, constraints (14) and (15) define the decision variables.

Proposition 1: If an optimal solution exists such that an EV leaves the depot with its battery partially charged, i.e. $Y_0^* < 1$, then the same EV departing from the depot fully charged is also optimal, i.e. $Y_0^* = 1$ is also optimal.

Proof: Let $Y_0^* < 1$ be optimal. Since fully recharging the battery at the depot does not delay the departure time of the EV $Y_0^* = 1$ must also be optimal.

Corollary 1: If $Y_0^* < 1$ is optimal, then the problem has infinite multiple optima.

Proof: Let $\bar{Y}_0 < 1$ be optimal and $\bar{Y}_0 + \varepsilon \leq 1$ not, where ε is a small positive scalar. Then following Proposition 1, multiple optima exist such that $\bar{Y}_0 \leq Y_0^* \leq 1$.

Proposition 2: If an optimal solution exists such that an EV has been recharged at least once and returns to the depot at the end of its route with positive battery state, i.e. $y_{n+1}^* > 0$, then its return to the depot with empty battery is also optimal, i.e. $y_{n+1}^* = 0$ is also optimal.

Proof: Let $y_{n+1}^* > 0$ be optimal. Since recharging the battery less at the preceding station does not delay the departure time to cause any time window infeasibility, $y_{n+1}^* = 0$ must also be optimal.

Corollary 2: If $y_{n+1}^* > 0$ is optimal, then the problem has infinite multiple optima.

Proof: Let $\bar{y}_{n+1} > 0$ be optimal and $\bar{y}_{n+1} + \varepsilon$ not, where ε is a small positive scalar. Then following Proposition 2, multiple optima exist such that $0 \leq y_{n+1}^* \leq \bar{y}_{n+1}$.

Without loss of generality, we assume that an EV departs from the depot with a battery charged in full and returns to the depot with its battery consumed if it has been recharged at least once along its route.

4. Solution methodology

We first construct a feasible initial solution using a heuristic similar to the insertion algorithm of Solomon (1987). We iteratively insert the customer which increases the total distance the least while satisfying time and battery charge constraints. If any customer cannot be inserted because of low battery level, then the candidate customer is inserted along with a recharging station. If no customer can be added to the current route due to capacity or time-window constraint, then we open a new route and continue the same procedure until all customers have been serviced. Then, the existing feasible solution is destroyed by removing some customers and recharging stations from their routes and then repaired by inserting the removed customers to the solution along with stations when recharging is necessary.

The proposed ALNS approach includes four classes of algorithms: Customer Removal (CR), Customer Insertion (CI), Station Removal (SR), and Station Insertion (SI). The removal and insertion algorithms are applied by selecting them dynamically and adaptively based on their past performances. The new solution is accepted according to the Simulated Annealing rule.

Algorithm 1: The general structure of the removal procedure

$\boldsymbol{input}$: Current feasible solution S_C, number of customers to be removed γ
$\boldsymbol{output}$: The set of customers which will be removed

Initialize removal list ($\mathcal{L} \leftarrow \emptyset$)
Apply a removal operator to find a set of customers for removal, ψ
$\mathcal{L} \leftarrow (\mathcal{L} \cup \psi)$
$\boldsymbol{Return}\ \mathcal{L}$

4.1. Removal algorithms

4.1.1. Customer removal

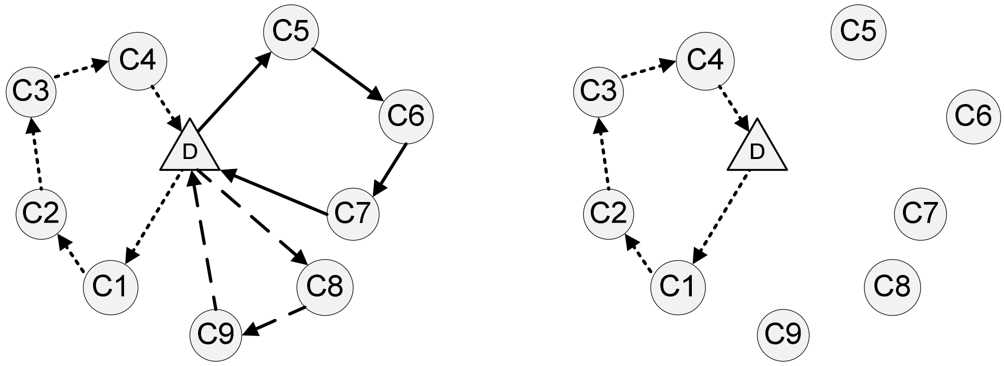
The current solution is destroyed by removing γ customers from the solution according to different rules and adding them in a removal list $\mathcal{L}$. γ depends on the total number of customers n_c and is determined randomly between $\underline{n_c}$ and $\overline{n_c}$ using a uniform distribution. The removal algorithms are selected in an adaptive manner from the set of algorithms CR. The general structure of the customer removal procedure is given in Algorithm 1.

We utilize Random, Worst-Distance, Worst-Time, Shaw, Proximity Based, Demand Based, Time Based, Zone Removal algorithms (see Demir et al. (2012) for details) and adopt the Route Removal algorithms presented in Emeç et al. (2014).

Random Route Removal (RRR): This algorithm randomly chooses ω routes and removes all the customers visited in those routes. ω depends on the number of routes in the current solution and is determined randomly between 10% and m_r% of total number of routes.

Greedy Route Removal (GRR): This algorithm removes ω routes in a greedy way. ω is determined in the same way as in RRR. The routes are sorted in the non-decreasing order of the number of customers serviced and ω routes are removed starting from the first route in the order.

The motivation is to distribute the customers in shorter routes into other existing routes in the solution in an attempt to reduce the number of vehicles. The procedure is illustrated in Fig. 2 for ω=2.



a) Current feasible solution b) Solution after GRR

Figure 2. Illustration of Greedy Route Removal

Note that we explicitly perform RRR and GRR after a predetermined number of iterations (N_{RR}) during a predetermined number of iterations (τ) to extensively attempt to reduce the number of vehicles used. RRR and GRR remove the complete routes from the solution. On the other hand, since other removal algorithms remove customers from their routes the battery state, time, and remaining capacity of the EV at its arrival to a node should be updated. Furthermore, some recharges may no longer be necessary and those stations may be removed from the solution. In fact, an EV may visit a recharging station right before or after servicing a customer, and it might be beneficial to remove the customer from the solution with its preceding or succeeding station. So, we introduce the following two operators for the customer removal algorithms in addition to removing customers only (RCO) option:

Remove Customer with Preceding Station (RCwPS): We remove the customer in the removal list along with the preceding station, if any exists. The idea is to eliminate the visit to a station where recharging is not necessarily needed at that battery state if EV no longer visits the removed customer.

Remove Customer with Succeeding Station (RCwSS): We remove the customer in the removal list along with the succeeding station, if any exists. The idea is similar to RCwPS. The recharging may be needed after departing from a customer in order to be able to reach the next customer in the route. In that case, recharging is not necessarily needed at that battery state if the departure customer is removed from the solution and the station can be removed as well.

Algorithm 2: Worst Charge Usage Station Removal

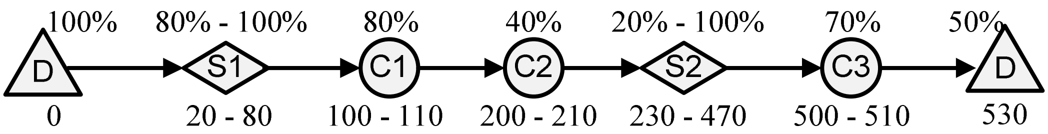
$\boldsymbol{input}$: Current feasible solution S_1, number of stations to be removed σ
$\boldsymbol{output}$: Destroyed solution S_2, Route list to be repaired $\mathcal{L}$

Initialize charge status hash map as $C \leftarrow \emptyset$
Initialize number of removed stations as $s \leftarrow 0$
$\boldsymbol{for}$ all stations in S_1
    Determine battery charge state of arriving EV
    Record the charge in C with station ID
$\boldsymbol{end\ for}$
Sort charge states in C in non-increasing order
$\boldsymbol{while}$ $s < \sigma$
    Select first element in C and identify station
    Remove that station from its route
    $s \leftarrow s + 1$
    $\boldsymbol{if}$ route number is not added in $\mathcal{L}$ before $\boldsymbol{then}$
        Add the route number to $\mathcal{L}$
$\boldsymbol{end\ while}$
$\boldsymbol{Return}$ S_2 *and* $\mathcal{L}$

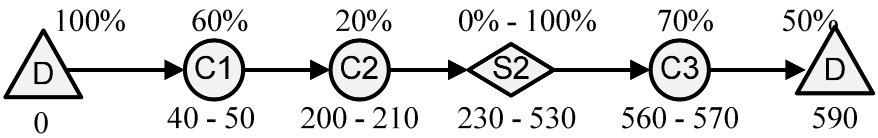
4.1.2. Station removal

The recharging stations are the crucial components of the problem. Hence, removing them or changing their positions in the visit sequence of a route may also improve the solution. So, after a pre-determined number of iterations, an SR (followed by a Station Insertion) procedure is applied. The number of stations to be removed σ is determined in a similar fashion to γ. The Random Station and Worst Distance Station Removal mechanisms are similar to their customer removal counterparts. We also use the Worst Charge Usage Station Removal which aims at removing the stations visited with high battery levels and Full Charge Station Removal which aims at promoting the PR option.

Worst Charge Usage Station Removal: The motivation of this algorithm is to make use of the battery as much as possible before a recharging is needed and increase the efficiency in using the stations. We promote the removal of the stations which an EV visits with relatively higher charge level. The stations are sorted in the non-increasing order of the battery level of the EVs that visit them for recharging and σ stations are removed starting from the first station in the order. The pseudocode is given in Algorithm 2.



a) Feasible route before SR

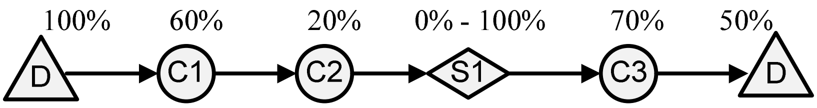


b) Time-window infeasible route after SR

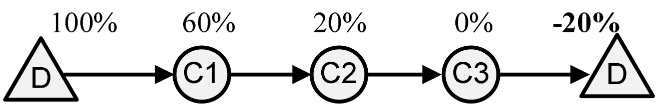
Figure 3. Example of time-window infeasibility after SR

Full Charge Station Removal: The algorithm identifies the stations where EVs are fully charged and removes γ of them randomly.

After the removal algorithms, the destroyed solution may become infeasible with respect to the state of charge. Consider the route shown in Fig. 3(a) as an example. The % numbers above the route indicate the state of charge at the arrival to and departure from a node whereas the numbers under the route show the arrival and departure times. When S1 is removed from the route, the EV can still visit C1 and C2 in the given sequence. However, since its battery is empty, the recharging takes longer at S2, which delays its arrival to C3. Since the EV departs from C3 at a later time, it cannot return to D before the latest arrival time of 550 as shown in Fig. 3(b).



a) Feasible route before SR



b) Battery infeasible route after SR

Figure 4. Example of battery infeasibility after SR

Fig. 4 illustrates how the battery infeasibility may occur after an SR. Consider the feasible route in Fig. 4(a). The EV is charged to full at S1. However, when S1 is removed, the battery level is not sufficient to return to the depot after visiting C3 as shown in Fig. 4(b).

4.2. Insertion algorithms

4.2.1. Customer insertion

We use the Greedy, Regret-2, and Regret-3 Insertion algorithms from the literature (see Demir et al. (2012) for details). In addition, we propose the Time Based Insertion and adapt the zone insertion as follows:

Time Based Insertion: In this algorithm, the insertion cost is calculated as the difference between the total route times before and after the insertion of a customer. For each customer, the algorithm determines the best insertion position among all routes based on this insertion cost. The customer which increases the route time the least is selected and inserted. The procedure is repeated for the remaining customers until all customers are inserted. The aim of this algorithm is to increase the number of customers visited by an EV by combining compatible customers with respect to their time windows or distances.

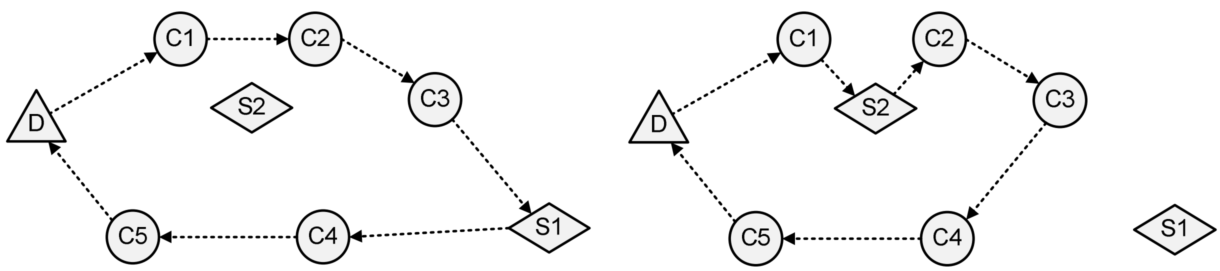
Zone Insertion: The algorithms uses the Time Based Insertion cirterion above when selecting a customer. However, instead of investigating all routes in the solution, it considers the routes within a randomly selected zone only.

To determine the battery state of charge and the recharge amount at a station visited in the implementation of CI algorithms we use the assumptions stated at the end of Section 3: an EV departs from the depot with a full battery and returns to the depot by completely consuming its battery if it has been recharged at least once along its route. So, in the case the EV is recharged only once during its route then: (i) if the customer is inserted between the depot and the station the insertion only affects the arrival state of charge at the station; (ii) if the customer is inserted between the station and the depot the recharge amount is increased such that the EV returns to the depot with empty battery.

If multiple recharges exist along the route and the customer is inserted between the depot and the first station visited, we follow the procedure (i) described above. If the customer is inserted between two consecutive stations or between the last station visited and the depot the amount recharged at the station is increased by the additional energy needed to visit that customer. If the recharge duration makes the insertion infeasible with respect to service time window of an existing customer, then we attempt to reduce the recharge duration by increasing the battery charge level at the arrival to that station. This is achieved by recharging the EV longer at the previous station making sure that the time-window feasibility of the customers visited between these two consecutive stations is maintained. In any case, if the insertion is feasible with respect to service time window but infeasible with respect to the battery state (referred to as charge infeasibility), the Greedy Station Insertion (see Section 4.2.2) is applied to make the destroyed solution charge feasible.

4.2.2. Station insertion

After removing some stations, the current feasible solution may become battery infeasible. In order to repair the solution, stations must be inserted to the infeasible routes. We make an infeasible route feasible by identifying the first customer at which the vehicle arrives with a negative battery level and inserting a station into the partial route prior to that customer. The difference from CI algorithms is that SI algorithms do not necessarily insert the stations which have been removed in SR. Since the stations are always available and it is assumed that as many stations as needed are available, any station can be inserted throughout the algorithm. The SR and SI procedure is illustrated on an example in Fig. 5. A feasible route is depicted in Fig. 5(a). Suppose, S1 is removed using an SR algorithm. Next, S2 is inserted between C1 and C2 by maintaining both time-window and battery feasibility. The resulting route in Fig. 5(b) is shorter than the initial.



a) Feasible solution b) Improved solution after SR and SI

Figure 5. An improved route after SR and SI procedure

We use the following three SI algorithms:

Greedy Station Insertion (GSI): This algorithm determines the first customer in the route at which the vehicle arrives with negative battery level and inserts the "best" (which increases the distance least) station on the arc between that customer and the previous customer. If this insertion is not feasible, then the previous arcs are attempted in the same manner.

Greedy Station Insertion with Comparison: The algorithm determines the best station on the arc leading to the customer where the battery level is negative as in GSI and compares the outcome with the case of inserting the corresponding best station on the immediate predecessor arc. The insertion which increases the route distance the least is performed. If both insertions are infeasible, the GSI procedure is applied by considering the previous arcs.

Best Station Insertion: We determine the best station insertions on all the arcs between the customer that the EV arrives with negative battery level and the depot or a previously visited station. We select the best feasible insertion and perform it.

The procedure is repeated for all customers where the EV arrives with negative battery level. If a station insertion cannot be performed feasibly, we return to the previous feasible solution.

Algorithm 3: ALNS algorithm

```
1    Generate an initial solution
2    j ← 1
3    while (stopping criterion is not satisfied) do
4        Select CR algorithm and remove customers
5        if destroyed solution infeasible then
6            Perform Greedy Station Insertion
7        Select CI algorithm and repair solution
8        Accept solution using SA criterion
9        j ← j + 1
10       if j ≡ 0 (mod N_RW1) then
11           Update adaptive weights of CR and CI algorithms
12       if j ≡ 0 (mod N_SR) then
13           Select SR algorithm and remove stations
14           Select SI algorithm and repair solution
15           Recall steps 8-10
16           if j ≡ 0 (mod N_RW2) then
17               Update adaptive weights of SR and SI algorithms
18   end while
```

The battery state of the EV and/or the recharge quantities in the implementation of SI algorithms are determined in a similar fashion to CI. We also implement a strategy where the recharge amount is constant and test different amounts for comparison purposes. In this case, we first try the constant PR option. Whenever the PR is insufficient to complete the route we switch from PR to FC at the last station visited before the infeasibility occurs.

The general structure of the ALNS algorithm is given in Algorithm 3.

5. Computational study

To validate the performance of the proposed ALNS approach we perform computational experiments using the EVRPTW data of Schneider et al. (2014). This data set was generated based on the well-known 56 VRPTW instances of Solomon (1987) and includes three main problem classes where 100 customers and 21 recharging stations are clustered (C), randomly distributed (R), and both clustered and randomly distributed (RC) over a 100×100 grid. Each set has also two subsets, type 1 and type 2, which differ by the length of the time windows and the vehicle capacity. We first tuned the parameters values using a subset of EVRPTW instances. Then, we solved the large EVRPTW problems and compare the results with the benchmarks reported in the literature. Finally, we report the solutions we achieved using the EVRPTW-PR setting and discuss the results. The algorithm was coded in the Java programming language.

5.1. Parameter tuning

We adopted a parameter tuning methodology similar to that of Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007), and Demir et al. (2012). We selected six large problems and performed ten runs for each parameter by considering reasonable values. At each step, we allowed one parameter to take a number of predefined values while the rest of the parameters were kept fixed. For each parameter, we selected the value that gave the least average deviation from the best achieved solution. After a parameter value has been determined, its value was fixed and this procedure was repeated for the remaining parameters until all parameters had been tuned. The details are given in Appendix A.

We omitted C1 and C2 problem classes since they usually converged to the same solutions for different parameter values and did not provide much information about the contribution of the parameter value on the solution quality. Consequently, we selected the instances R107, RC101, RC104, RC105, R205 and RC205 for parameter tuning.

We observed that if the score of the worse solution (σ_3) is greater than the score of the better solution (σ_2) it allows diversification by rewarding non-improved solutions as noted in Ropke and Pisinger (2006a) and Demir et al. (2012). So, our setting of the parameters σ_1, σ_2, and σ_3 also rewards a worse solution more than a better solution as follows: $\sigma_1 \geq \sigma_3 \geq \sigma_2$.

Ropke and Pisinger (2006a) set the number of iterations to 25,000 and noted that additional runtime had a minor contribution to the solution quality. Our convergence analysis showed similar results. So, we also set the number of iterations to 25,000.

5.2. Numerical results for EVRPTW instances

In Table 1 we compare our average results and the results reported in Goeke and Schneider (2015) and Hiermann et al. (2015) to those given in Schneider et al. (2014). All these results were obtained with fixed parameter values. The first column denotes the instances. '*#Veh*' refers to the number of vehicles and '$\Delta\%$' is the percentage deviation from the distances reported in Schneider et al (2014), if the number of vehicles is same. A negative $\Delta\%$ value means improvement.

Table 1. Average results for EVRPTW obtained with fixed parameters

Instance	SSG		GS		HPH		KÇ	
Type	*#Veh*	*TD*	*#Veh*	Δ%	*#Veh*	Δ%	*#Veh*	Δ%
c1	10.67	1050.04	10.67	-0.31	10.67	0.16	10.89	0.78
c2	4.00	640.92	4.00	0.00	4.00	0.06	4.00	0.00
r1	12.83	1268.60	12.83	-0.80	13.00	0.11	13.25	0.69
r2	2.64	919.04	2.64	-0.47	2.64	0.53	2.82	-0.07
rc1	13.13	1415.84	13.13	-0.50	13.00	-0.48	13.38	0.13
rc2	3.13	1146.76	3.13	-0.15	3.13	0.95	3.25	0.08
Average				-0.41		0.25		0.25

SSG: Schneider et al. (2014), GS: Goeke and Schneider (2015), HPH: Hiermann et al. (2015), and KÇ: Our ALNS

Our results show that our ALNS approach performs better in type-2 problems but it converges to solutions with one additional vehicle in several instances as compared to other methods. Even though the performance of Hiermann et al. (2015) is better in type-1 problems, the overall performance of our approach is similar. We also observe that the recent work of Goeke and Schneider (2015) has a superior performance, improving many of the best solutions to date. We note that Goeke and Schneider (2015) used the numbers of vehicles achieved in Schneider et al. (2014) as a priori information to construct their initial routes, which might have a positive effect both in run time and solution quality. Nevertheless, with fixed parameters our approach improved the best solutions of 11 instances.

In Table 2, we provide our best results along with the best known solutions (BKS) reported in the literature. '*TD*' refers to the total distance. These results also show that our ALNS approach performs better in type-2 problems. Although it finds solutions with one additional vehicle in some instances, it improved the best known solutions of four instances, of which three are type-1 problems. Furthermore, it achieved the same best known solution in 16 instances. In comparison, Schneider et al. (2014) found the best known solutions in 18 instances whereas Hiermann et al. (2105) and Goeke and Schneider (2015) improved the solutions of 4 and 30 instances, respectively.

Table 2. Comparison with the best known solutions of EVRPTW instances

	BKS			KÇ				BKS			KÇ		
Inst.	*#Veh*	*TD*	Ref.	*#Veh*	*TD*	Δ%	Inst.	*#Veh*	*TD*	Ref.	*#Veh*	*TD*	Δ%
c101	12	1053.83	SSG	12	1053.83	0.00	c201	4	645.16	SSG	4	645.16	0.00
c102	11	1051.38	GS	11	1056.12	0.45	c202	4	645.16	SSG	4	645.16	0.00
c103	10	1034.86	GS	11	1001.81	-	c203	4	644.98	SSG	4	644.98	0.00
c104	10	961.88	GS	10	**951.57**	**-1.08**	c204	4	636.43	SSG	4	636.43	0.00
c105	11	1075.37	SSG	11	1075.37	0.00	c205	4	641.13	SSG	4	641.13	0.00
c106	11	1057.65	HPH	11	1057.65	0.00	c206	4	638.17	SSG	4	638.17	0.00
c107	11	1031.56	SSG	11	1031.56	0.00	c207	4	638.17	SSG	4	638.17	0.00
c108	10	1095.66	GS	11	1015.68	-	c208	4	638.17	SSG	4	638.17	0.00
c109	10	1033.67	GS	10	1069.16	3.32							
r101	18	1663.04	HPH	18	1679.06	0.95	r201	3	1264.82	SSG	3	1265.67	0.07
r102	16	1487.41	GS	16	1524.14	2.41	r202	3	1052.32	SSG	3	1052.32	0.00
r103	13	1271.35	GS	13	1312.50	3.14	r203	3	895.54	GS	3	895.54	0.00
r104	11	1088.43	SSG	12	1071.89	-	r204	2	779.49	GS	2	780.98	0.19
r105	14	1442.35	GS	15	1383.29	-	r205	3	987.36	GS	3	987.36	0.00
r106	13	1324.10	GS	14	1276.15	-	r206	3	922.19	GS	3	922.70	0.06
r107	12	1150.95	GS	12	**1148.43**	**-0.22**	r207	2	845.26	GS	2	847.14	0.22
r108	11	1050.04	SSG	11	1051.59	0.15	r208	2	736.12	GS	2	736.12	0.00
r109	12	1261.31	GS	13	1214.72	-	r209	3	867.05	GS	3	871.22	0.48
r110	11	1119.50	GS	12	1097.89	-	r210	3	846.20	GS	3	**843.65**	**-0.30**
r111	12	1106.19	SSG	12	1109.14	0.27	r211	2	827.89	GS	3	761.56	-
r112	11	1016.63	GS	11	1038.74	2.13							
rc101	16	1726.91	HPH	16	1731.07	0.24	rc201	4	1444.94	SSG	4	1446.84	0.13
rc102	14	1552.08	HPH	15	1551.69	-	rc202	3	1410.74	GS	3	1450.34	2.73
rc103	13	1350.09	GS	13	1351.73	0.12	rc203	3	1055.19	GS	3	1069.27	1.32
rc104	11	1227.25	GS	11	1232.45	0.42	rc204	3	884.80	GS	3	887.45	0.30
rc105	14	1475.31	HPH	14	**1473.24**	**-0.14**	rc205	3	1273.55	GS	3	1277.60	0.32
rc106	13	1427.21	GS	14	1414.99	-	rc206	3	1188.63	GS	3	1207.64	1.57
rc107	12	1274.89	SSG	12	1283.05	0.64	rc207	3	985.03	GS	3	994.48	0.95
rc108	11	1197.83	GS	11	1209.11	0.93	rc208	3	836.29	GS	3	841.34	0.60
Avg	12.21			12.52		0.69		3.19			3.22		0.33

SSG: Schneider et al. (2014), GS: Goeke and Schneider (2015), HPH: Hiermann et al. (2015), and KÇ: Our method

With the respect to computational effort, Schneider et al. (2014) reported an average run time of 15.34 minutes on an Intel Core i5 processor with 2.67 GHz speed and 4 GB RAM, operating on Windows 7 Professional. Goeke and Schneider (2015) uses an Intel Core i7 processor at 2.8GHz, 8 GB of RAM running on Windows7 Enterprise and reported an average run time of 2.78 minutes whereas Hiermann et al. (2105) run their experiments on a single core of a cluster system with an Intel Core2 Quad CPU Q6600 with 2.40 GHz, 4 GB RAM, operating on 64-Bit

Linux operating system and achieves an average run time of 15.92 minutes. Our average run time is 12.26 minutes on an Intel Xeon E5 processor with 3.30 GHz speed and 32 GB RAM using 64-bit Windows 7 operating system.

5.3. Numerical results for EVRPTW-PR instances

For the EVRPTW-PR, we tested two different versions of our ALNS method. In the first, the recharge quantity q can take any value whereas in the second it is set equal to a pre-determined constant. For performance comparison we tested the cases where q=0.3, q=0.4, and q=0.5. The results are reported in Table 4. These results show the advantage of using the PR strategy over the full charging restriction. We observe that the average distances improve significantly in type-1 instances, c1 problem set in particular. Furthermore, we note that the number of vehicles is reduced by one vehicle in some instances, mostly in type-1 problems. As expected, ALNS using the variable recharge quantity has a better performance. Nevertheless, ALNS with the constant recharge quantity is also capable of obtaining high quality solution. Overall, these results suggest the PR scheme is effective, particularly in cases where the time windows are more restrictive.

6. Conclusions and future research

In this paper, we investigated the partial recharge strategies for the EVRPTW, namely EVRPTW-PR, and proposed an ALNS algorithm to solve it. Some of the existing ALNS mechanisms were adopted from the literature whereas new removal and insertion mechanisms specific to EVRPTW were developed to handle the visits to recharging stations and to incorporate the PR decisions.

We used the instances generated by Schneider et al. (2014) to validate the performance of the proposed ALNS. We first solved the EVRPTW instances and benchmarked our results with those of reported in Schneider et al. (2014), Goeke and Schneider (2015), and Hiermann et al. (2015). We also reported 4 new best-known solutions. For the proposed EVRPTW-PR we solved the same instances. The results revealed that the routes can be significantly improved when PR is allowed, even if at a pre-determined constant level.

Table 4. EVRPTW-PR results for different recharge strategies

	FC		PR (q free)		PR (q=0.3)		PR (q=0.4)		PR (q=0.5)	
Inst.	*#Veh*	*TD*	*#Veh*	Δ%	*#Veh*	Δ%	*#Veh*	Δ%	*#Veh*	Δ%
c101	12	1053.83	12	-0.25	12	-1.00	12	-0.75	12	0.00
c102	11	1056.12	11	-1.71	11	-2.09	11	-2.29	11	-1.52
c103	11	1001.81	10	-	10	-	10	-	10	-
c104	10	951.57	10	-7.31	10	-6.79	10	-6.58	10	-6.44
c105	11	1075.37	11	-3.62	11	-2.22	11	-2.22	11	-1.99
c106	11	1057.65	11	-3.27	11	-2.64	11	-2.00	11	-1.06
c107	11	1031.56	11	-1.11	11	-1.69	11	-1.77	11	-1.77
c108	11	1015.68	11	-0.95	11	-1.61	11	-1.31	11	-1.51
c109	10	1069.16	10	-11.37	10	-7.94	10	-12.88	10	-12.92
c201	4	645.16	4	-2.41	4	-0.36	4	-0.36	4	-0.63
c202	4	645.16	4	-2.41	4	-0.64	4	-0.64	4	-0.64
c203	4	644.98	4	-2.39	4	-0.61	4	-0.61	4	-0.61
c204	4	636.43	4	-1.03	4	-0.10	4	0.00	4	0.00
c205	4	641.13	4	-1.77	4	-0.46	4	-1.56	4	-0.46
c206	4	638.17	4	-1.30	4	0.46	4	-1.09	4	0.00
c207	4	638.17	4	-1.30	4	0.46	4	0.00	4	0.00
c208	4	638.17	4	-1.30	4	-0.63	4	-1.09	4	0.00
r101	18	1679.06	18	-0.68	18	-2.59	18	-1.65	18	0.01
r102	16	1524.14	16	-4.29	16	-3.92	16	-4.29	16	-3.83
r103	13	1312.50	14	-	14	-	13	-3.68	13	-2.80
r104	12	1071.89	12	-0.62	11	-	12	-0.63	12	-0.63
r105	15	1383.29	15	-0.68	15	-0.19	15	-0.21	15	0.97
r106	14	1276.15	13	-	14	0.99	14	1.50	14	0.63
r107	12	1148.43	12	-2.64	12	-1.42	12	-1.54	12	-1.95
r108	11	1051.59	11	-1.98	11	-0.54	11	-0.65	11	-0.30
r109	13	1214.72	13	-1.02	13	-1.76	13	-0.45	13	-1.67
r110	12	1097.89	12	-0.67	11	-	12	-0.17	12	-0.38
r111	12	1109.14	12	-2.31	12	-0.64	12	-1.18	12	-0.71
r112	11	1038.74	11	-2.11	11	-0.35	11	-2.09	11	-0.08
r201	3	1265.67	3	-8.38	3	0.07	3	0.65	3	-0.28
r202	3	1052.32	3	0.00	3	0.23	3	0.12	3	0.30
r203	3	895.54	3	0.00	3	0.13	3	0.03	3	0.02
r204	2	780.98	2	-0.11	3	-	3	-	2	2.53
r205	3	987.36	3	0.00	3	0.17	3	0.77	3	0.42
r206	3	922.70	3	0.00	3	0.29	3	1.30	3	0.32
r207	2	847.14	2	-0.06	2	0.70	3	-	2	0.28
r208	2	736.12	2	0.00	2	0.09	2	0.09	2	0.17
r209	3	871.22	3	-0.26	3	0.01	3	0.09	3	0.04
r210	3	843.65	3	-0.03	3	0.84	3	0.65	3	0.77
r211	3	761.56	2	-	3	0.00	3	0.65	3	0.65
rc101	16	1731.07	16	-2.38	15	-	16	-2.47	16	-1.58
rc102	15	1551.69	15	-1.40	14	-	14	-	14	-
rc103	13	1351.73	13	-1.67	13	-0.02	13	-0.69	13	-0.50
rc104	11	1232.45	11	-2.45	11	-0.50	11	-0.25	11	-0.71
rc105	14	1473.24	14	-1.01	14	-1.03	14	-0.40	14	-0.19
rc106	14	1414.99	13	-	13	-	13	-	13	-
rc107	12	1283.05	12	-1.75	12	-0.86	12	-1.41	12	-1.71
rc108	11	1209.11	11	-1.98	11	-2.12	11	2.16	11	0.04
rc201	4	1446.84	4	0.00	4	0.50	4	1.15	4	0.63
rc202	3	1450.34	3	-2.36	4	-	4	-	4	-
rc203	3	1069.27	3	0.00	3	1.88	3	2.10	3	1.36
rc204	3	887.45	3	0.04	3	-0.02	3	0.54	3	-0.14
rc205	3	1277.60	3	-1.22	4	-	4	-	4	-
rc206	3	1207.64	3	0.51	3	-0.13	3	0.06	3	0.15
rc207	3	994.48	3	-0.10	3	0.19	3	-0.24	3	-0.12
rc208	3	841.34	3	-0.19	3	0.76	3	0.31	3	0.40
Avg				-1.67		-0.81		-0.92		-0.73

In this study, we only allowed PR using the same power level. The problem can be extended to a multiple recharge power options at different speeds and costs as discussed in Felipe et al. (2014). Further research on this topic may also address the heterogeneous fleet case. The heterogeneity within this context does not only arise from the vehicle capacities but from their batteries as well since the cruising range of EVs and discharge/recharge durations differ depending on their battery condition and age.

Acknowledgments

This research was partially supported by The Scientific and Technical Research Council of Turkey Grant #113M522 and BIDEB Program scholarship to the first author.

Appendix A: Parameter tuning

The parameters are described in Table A.1. The values considered for tuning are given in Table A.2. For each parameter, the first line is the value and the second shows the deviation between the average solution achieved with that value and the best solution found in all runs. We reported the initial parameter values in the first column. The tuning sequence is from top to bottom.The best parameter values (the ones yielding the minimum deviations) are indicated in bold.

Table A.1. Notation and description of the parameters

σ_2	Score of the better solution
N_{RW1}	# of iterations between which adaptive weights are updated (for CR and CI algorithms)
ρ	Roulette wheel parameter
σ_1	Score of the best solution
σ_3	Score of the worse solution
ϕ_1	First Shaw parameter
ϕ_2	Second Shaw parameter
ϕ_3	Third Shaw parameter
ϕ_4	Fourth Shaw parameter
ε	Cooling rate of SA
μ	Start temperature control parameter of SA
κ	Worst removal determinism factor
η	Shaw removal determinism parameter
n_Z	Number of zones (Zone removal)
N_{SR}	# of iterations between which SR is performed
N_{RW2}	# of iterations between which adaptive weights are updated (for SR and SI algorithms)
m_r	Route removal upper bound
N_{RR}	# of iterations between which route removal algorithms are performed
τ	Consecutive # of iterations during which route removal algorithms are performed

Table A.2. Parameter values and the corresponding deviations

Parameters	Parameter Values and Corresponding Deviations										
σ_2	Value	**6**	0	2	4	9	12	14	16	18	20
	Deviation	**0.33**	0.42	0.42	0.43	0.40	0.42	0.42	0.37	0.35	0.42
N_{RW1}	Value	300	50	100	150	200	250	350	400	450	**500**
	Deviation	0.38	0.55	0.52	0.40	0.40	0.37	0.38	0.38	0.38	**0.32**
ρ	Value	0.35	0.05	0.1	0.15	**0.2**	0.25	0.3	0.4	0.45	0.5
	Deviation	0.60	0.55	0.62	0.60	**0.50**	0.53	0.58	0.53	0.57	0.58
σ_1	Value	**33**	5	10	20	25	30	35	40	45	50
	Deviation	**0.50**	0.57	0.50	0.55	0.53	0.58	0.58	0.57	0.53	0.53
σ_3	Value	3	6	9	12	13	15	**21**	24	27	30
	Deviation	0.53	0.55	0.55	0.57	0.58	0.55	**0.50**	0.62	0.55	0.57
ϕ_1	Value	0.5	1	3	**5**	7	9	11	13	15	
	Deviation	0.55	0.53	0.58	**0.50**	0.60	0.57	0.52	0.52	0.65	
ϕ_2	Value	9	0.25	**1**	3	5	7	11	13	15	
	Deviation	0.55	0.48	**0.48**	0.55	0.55	0.53	0.55	0.62	0.55	
ϕ_3	Value	11	0.15	1	3	5	7	9	**13**	15	
	Deviation	0.52	0.57	0.50	0.50	0.53	0.55	0.50	**0.48**	0.53	
ϕ_4	Value	8	**0.25**	1	2	3	4	5	6	7	9
	Deviation	0.53	**0.52**	0.58	0.62	0.60	0.52	0.55	0.57	0.52	0.55
ε	Value	0.9996	0.999	0.9991	0.9992	0.9993	0.9994	**0.9995**	0.9997	0.9998	0.9999
	Deviation	0.53	0.52	0.55	0.50	0.57	0.55	**0.48**	0.52	0.55	0.58
μ	Value	0.4	**0.05**	0.1	0.15	0.2	0.25	0.3	0.35	0.45	0.5
	Deviation	0.58	**0.50**	0.52	0.60	0.52	0.58	0.62	0.60	0.63	0.63
κ	Value	6	**1**	2	3	4	5				
	Deviation	0.58	**0.48**	0.53	0.53	0.53	0.50				
η	Value	12	2	4	6	8	**10**				
	Deviation	0.55	0.57	0.57	0.58	0.53	**0.48**				
n_Z	Value	15	5	7	9	11	13	19	**21**	25	30
	Deviation	0.52	0.60	0.55	0.58	0.57	0.60	0.53	**0.52**	0.57	0.60
N_{SR}	Value	50	**10**	20	30	40	60	70	80	90	100
	Deviation	0.58	**0.52**	0.62	0.55	0.55	0.52	0.58	0.57	0.58	0.62
N_{RW2}	Value	3500	**1000**	1500	2000	2500	3000	4000	4500	5000	5500
	Deviation	0.58	**0.50**	0.58	0.60	0.53	0.60	0.60	0.62	0.52	0.53
m_r	Value	**0.4**	0.3	0.5	0.6						
	Deviation	**0.50**	0.55	0.52	0.57						
N_{RR}	Value	5000	2000	2500	3000	3500	4000	4500	5500	**6000**	6500
	Deviation	0.60	0.55	0.63	0.52	0.58	0.58	0.58	0.57	**0.52**	0.55
τ	Value	**1000**	750	1250	1500	1750	2000	2250	2500	2750	3000
	Deviation	**0.47**	0.58	0.52	0.50	0.50	0.55	0.53	0.55	0.57	0.58

References

Adulyasak, Y., Cordeau, J-F., Jans, R. (2014). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science* 48(1):20-45

Aksen, D., Kaya, O., Salman, S., Tüncel, Ö. (2014). An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research* 239 (2):413-426.

Artmeier, A., Haselmayr, J., Leucker, M. and Sachenbacher, M. (2010). The optimal routing problem in the context of battery-powered electric vehicles. *2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability,* Bologna, Italy.

Azi, N., Gendreau, M., Potvin, J-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research 41*:167–173.

Bramel, J., Simchi-Levi, D., (1995). A location based heuristic for general routing problems. *Operations Research*, 43(4): 649-660.

Braysy, O., Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39(1): 104-118

Chan, C. C. (2002). The state of the art of electric and hybrid vehicles. *Proceedings of the IEEE,* 90(2): 247-275.

Chen, J. F., Wu, T. H., (2006). Vehicle routing problem with simultaneous deliveries and pickups. *The Journal of the Operational Research Society,* 57(5): 579-587.

Conrad, R. G., Figliozzi, M. A. (2011). The recharging vehicle routing problem. In: Doolen, T. and Van Aken, E. (eds.) *Proceedings of the 2011 Industrial Engineering Research Conference.*

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.

Desaulniers, G., Errico, F., Irnich, S., Schneider, M. (2014). Exact algorithms for electric vehicle routing problems with time windows. Working Paper.

Emeç U., Çatay B., Bozkaya B. (2014). An adaptive large neighborhood search for an e-grocery delivery routing problem. Working Paper, Sabanci University.

Erdogan, S., Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E* 48: 100–114.

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* 71: 111–128.

Freight Facts and Figures 2013 (2014). U.S. Department of Transportation, Bureau of Transportation Statistics.

Goeke, D., Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research* (in press).

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. (2015). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Working Paper.

Ioannou, M., Kritikos, M., Prastacos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows, *The Journal of the Operational Research Society* 52(5): 523-537.

Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7: 285-300.

Masson, R., Lehuédé, F,. Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transportation Science 47(3):344-355.

Muller, L.F. (2009). An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. *Proceedings of the VIII Metaheuristics International Conference* (MIC) *2009* (Hamburg, Germany).

Muller, L.F., Spoorendonk, S., Pisinger, D. (2012). A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *European Journal of Operational Research* 218(3): 614-623.

Omidvar, A., Tavakkoli-Moghaddam, R. (2012). Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. In *IEEE International Energy Conference and Exhibition, Florence,* 1089–1094

Pisinger, D., Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research* 34(8): 2403-2435.

Pisinger, D., Ropke, S. (2010). Large neighborhood search. In: Gendreau M, Potvin J-Y, (eds.) *Handbook of Metaheuristics* 146: 399-419.

Qu, Y., Bard, J.F. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research* 39(10):2439–2456

Ribeiro, G.M., Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research* 39(3): 728-735.

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.

Schneider, M. Stenger, A., Goeke, D. (2014). The electric vehicle routing problem with time windows and recharging stations. *Transportation Science* (forthcoming).

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G. (2000) Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* 159(2): 139-171.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming,* Springer, New York: 417-431.

Solomon, M. M., (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2): 254-265.

Toth, P., Vigo, D. (2002). An overview of vehicle routing problems. In: Toth P, Vigo D, (eds.) *The Vehicle Routing Problem,* SIAM, Philadelphia: 1-26.

Touati-Moungla, N., Jost, V. (2012). Combinatorial optimization for electric vehicles management. *Journal of Energy and Power Engineering* 6(5): 738-743.

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

Wang, H. and Cheu, R. L. (2012). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations, *Journal of the Transportation Research Board* 2352: 1-10

White Paper on Transport (2011). *Roadmap to a single European transport area: Towards a competitive and resource-efficient transport system*, Publications Office of the European Union, Luxembourg.

Worley, O., Klabjan, D. (2012). Simultaneous vehicle routing and charging station siting for commercial electric vehicles. *In IEEE International Electric Vehicle Conference,* Greenville, SC: 1-3.

http://ec.europa.eu/transport/strategies/facts-and-figures/all-themes/index_en.htm (last accessed on May 30, 2015)

http://www.epa.gov/climatechange/ghgemissions/sources/transportation.html (last accessed on May 30, 2015)

http://www.state.gov/e/oes/rls/rpts/car6/index.htm (last accessed on May 30, 2015)