Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis International Bachelor Econometrics and Operations Research

State of Charge-Induced Battery Deterioration in the Electric Vehicle Routing Problem with Time Windows and Partial Recharge Strategies

Egle Sakalauskaite (621819)

Supervisor:	BTC van Rossum
Second assessor:	Name of your second assessor
Date final version:	17th May 2024

Abstract

Your abstract

1 Introduction

the global sales of electric vehicles (EVs) have increased from 320 thousand in 2014 to 14 million in 2023, rapidly replacing conventional vehicles by reaching a market share of 16% in the previous year (Figure 1). It is estimated that by the year 2045, all newly sold vehicles will be zero-emission (Irle, 2024). This trend will continue to positively impact climate change; According to European Environment Agency (EEA, 2024), as of 2022, domestic transportation has still contributed 23.7% of all greenhouse gas emissions in EU. However, several challenges are faced when using electric vehicles: the limitations in battery capacity have made the driving range of EVs shorter than conventional vehicles, and in many areas, finding a charging station at reasonable proximity is a challenge. Additionally, full recharging takes a few hours, which is much longer than a gas refill (Pasha et al., 2024). Regardless, we should not forget that the advantages of using battery-powered vehicles significantly outweigh the drawbacks. Therefore, it is crucial to analyze the negative aspects to understand their impacts thoroughly: this will help address these challenges more effectively and improve the overall efficiency and sustainability of using EVs.

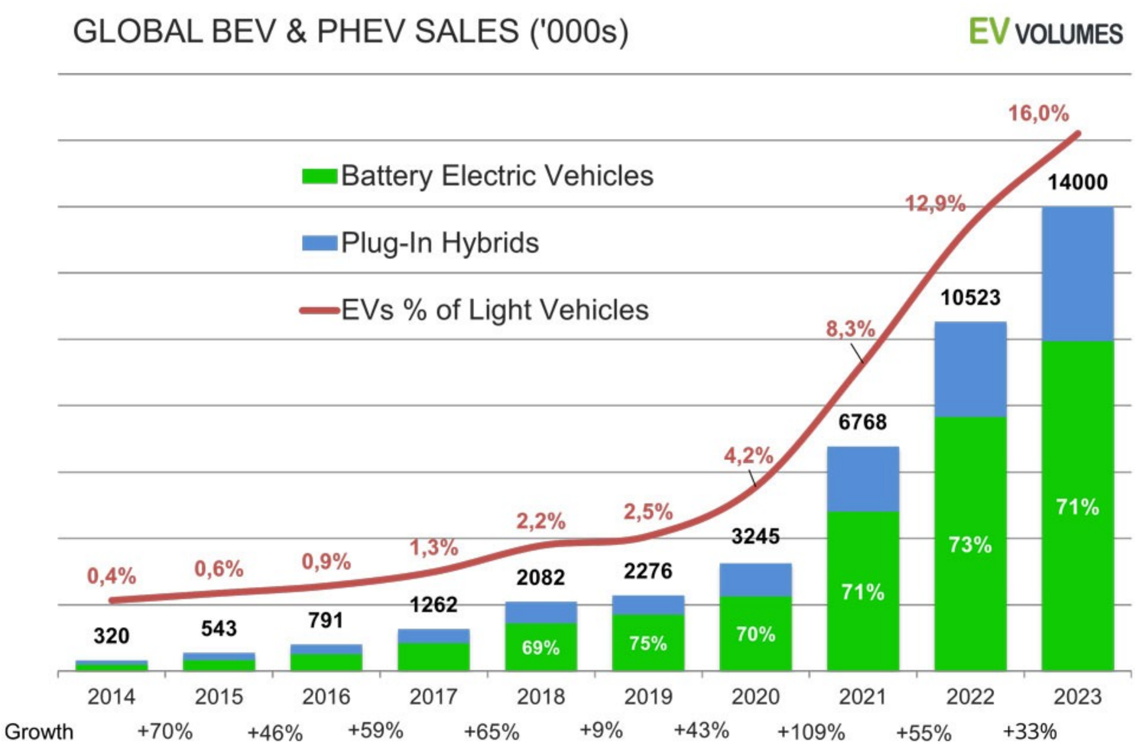

Figure 1: Global electric vehicle sales 2014-2023 (Irle, 2024)

Electric Vehicle Routing Problem (EVRP) is an extension of well-known *Vehicle Routing Problem* (VRP), in which EVs are used instead of traditional combustion engine vehicles. The goal is to determine the most efficient routes for a fleet of electric vehicles to serve several customers. In most cases, the efficiency of the routes is measured by the total driving distance and/or duration, but other factors may be regarded, such as the total number of vehicles used. Recharging or battery swapping is a crucial element in such problems, which presents an extra level of complexity than VRP.

In this paper, we explore the variant of the EVRP presented by Keskin and Çatay (2016), known as the *Electric Vehicle Routing Problem with Time Windows and Partial Recharge* (EVRPTW-PR). The objective is to route a homogeneous fleet of EVs while minimizing the total driving distance. In addition to the usual EVRP constraints, such as vehicle loading and battery capacity, customer service is restricted by delivery time windows. Since the charging duration is proportional to the battery level increase, allowing partial recharging improves flexibility in visiting customers before their deadlines. Consequently, the authors propose that vehicles are fully charged when leaving the depot and if a vehicle is recharged at least once during its route, it returns to the depot with the battery empty, as any additional recharging is time-inefficient.

However, there is scientific proof that allowing the battery to (nearly) fully discharge and charging it to full capacity can negatively affect its performance (BU, 2023). Figure 2 illustrates how Using the battery at a state of charge (SoC) between 65 and 75% (in orange) preserves its capacity the most. However, this is not very practical, especially considering EVs, since it would significantly reduce the already limited driving distance between charges. The best trade-off between preservation of the battery capacity and autonomous use seems to be the strategy that keeps the battery SoC between 25 and 85% (in green). According to König et al. (2021) batteries contribute to a large share of the total price of EV. Therefore, failing to preserve it can lead to huge financial losses.

EVRPTW-PR can be modeled using a mixed integer linear program (MILP). This problem is NP-hard. Therefore, solving the problem quickly becomes computationally challenging as the problem size increases. To overcome this, Keskin and Çatay (2016) proposed an *Adaptive Large Neighborhood Search* (ALNS) algorithm, which iteratively destroys the current solution by removing several locations and then repairs it by adding stations and the removed customers back to the solution in (different) positions. Various operations perform both the removal and insertion of customers; the algorithm tracks each of their performances, prioritizing the best-performing ones in future iterations.

The goal of this paper is to learn more about EVRPTW-PR by replicating the results of Keskin and Çatay (2016), as well as expanding on their research by introducing battery deterioration costs and examining the impact battery deterioration has on this problem's solution quality. We will aim to answer these following questions:

- Can we replicate the results of Keskin and Çatay (2016)? How do our results differ from theirs and what could be the cause?
- How does introduction of battery deterioration affect the costs of routing electric vehicles? What is the magnitude of financial loss when the solution is obtained while battery degradation is ignored?
- Can the ALNS approach find good feasible solutions if the battery State of Charge is

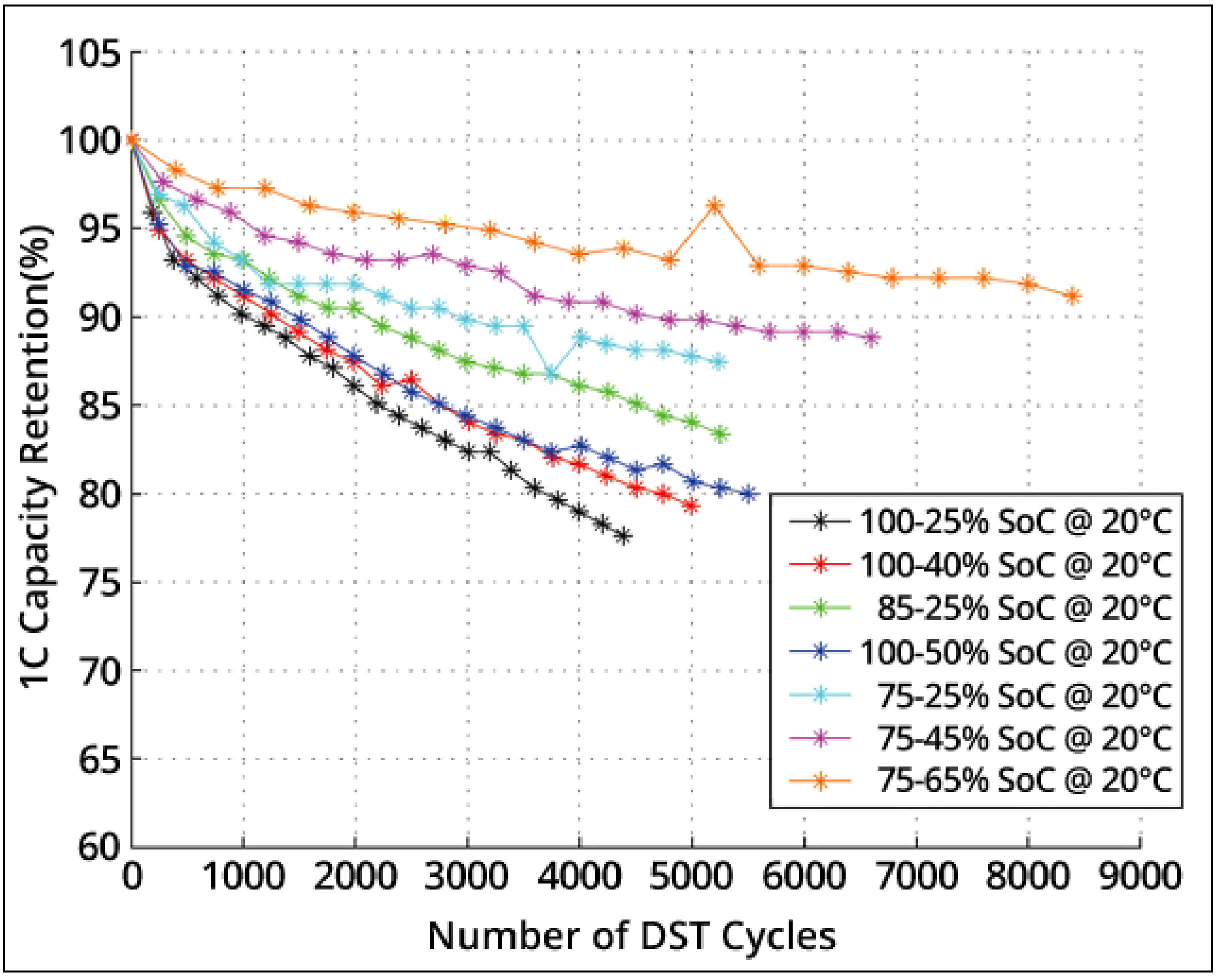

Figure 2: Capacity loss as a function of charge and discharge levels (BU, 2023)

restricted to avoid battery deterioration? Can the saved costs in battery deterioration when applying this restriction compensate for the larger driving distance?

This paper is organized as follows: First, Section 2 gives an overview of the development of EVRP and some existing problem extensions. Second, Section 3 describes EVRPTW-PR in detail by presenting a MILP formulation and extends it by including penalties for violating the recommended SoC. Next, Section 4 presents the algorithm of ALNS together with all insertion and removal operations it uses in this paper. Section 5 compares our findings with the original paper and the difference in results once the effects of SoC are considered. Finally, Section 6 gives a summary of our findings and discusses future research opportunities.

2 Literature review

The *Vehicle Routing Problem* (VRP) was first proposed by Dantzig and Ramser (1959). VRP focuses on finding the shortest route while delivering goods to numerous customers by a fleet of vehicles. Since then, many variations of the problem have been explored, incorporating different constraints that can better reflect the complexity of cases in real life. Particularly, Solomon (1987) has addressed the scheduling restrictions in VRP when the deliveries could only be performed during specific time windows. leading to the introduction of *Vehicle Routing Problem with Time Windows* (VRPTW).

By the late 20th century, concerns about climate change grew and began to be reflected in operations research studies that followed. Conrad and Figliozzi (2011) presented us with a *Recharging Vehicle Routing Problem* (RVRP), in which vehicles are allowed to recharge at the customer locations. Their approach not only minimized the total costs but also the number of vehicles required. Just as in Keskin and Çatay (2016), the delivery time window restrictions were also considered, and partial recharge was optional, but only to a fixed level, such as up to 80% of the battery's capacity. However, in practical scenarios, vehicle recharging is typically conducted at locations other than customer sites. A good implementation of this can be found in ~~a publication of~~ Wang and Cheu (2013), in which operations of the EV taxi fleet were explored, focusing only on the pre-booked trips. The authors used a two-phase heuristic approach by first constructing an initial solution using one of the nearest-neighbor, sweep, and earliest-time window insertion heuristics and refining it using tabu search. Additionally, they proposed three distinct recharging strategies, each offering different driving ranges, and evaluated these against a full charging scheme.

The study of Schneider, Stenger and Goeke (2014) introduced the *Electric Vehicle Routing Problem with Time Windows* (EVPTW) to address new environmental regulations that promote using electric vehicles for last-mile deliveries. To solve this problem, the authors developed a hybrid heuristic combining variable neighborhood search with tabu search techniques. It is evident that this article was an inspiration for Keskin and Çatay (2016). However, there was one key difference: Schneider et al. (2014) did not address the possibility of partial recharging. Since charging EVs takes time, partial recharging could be the key to more efficient schedules. This idea is backed up by Felipe, Ortuño, Righini and Tirado (2014). In their research, the authors explored a variation of a *Green Vehicle Routing Problem* (GVRP) with electric vehicles. The study considers not only the optimal routing schedule but also the specific amounts of energy to be recharged and the charging technology employed. A constructive and deterministic local search heuristics integrated within a non-deterministic Simulated Annealing framework was proposed. It was demonstrated that partial recharges can lead to considerable cost and energy savings and assist in maintaining feasibility in more complex scenarios.

After publishing Keskin and Çatay (2016), the authors continued their research on EVRPTW with the ALNS approach by exploring a different challenge that electric vehicles face. Due to long charging durations, some vehicles must wait at the recharging stations to use the facility (Keskin, Laporte & Çatay, 2019). This may introduce additional costs if driver salary is involved and affect the routes' feasibility when delivery time windows are present. Since these waiting times are dynamic in the real world, the time window constraints are relaxed to allow late arrivals. However, these scenarios result in additional penalty costs. Another drawback of electric vehicles is that their energy consumption rate fluctuates with the load they carry (Rastani & Çatay, 2021). According to the article, if the effects of load weight are ignored, the total operating costs might increase to up to 31%. Therefore, incorporating these concerns regarding electric vehicles into the EVRP formulations allows a more accurate representation of real-life problems and better results.

To the best of our knowledge, little research has been done regarding the effects of SoC on batteries in EVRP, but some studies explore this. Cataldo-Díaz, Linfati and Escobar (2022) incorporate this by first restricting the battery depletion below 25% of its capacity. Later, charging the battery to the full capacity is also constrained, allowing the vehicles only to maintain their battery in the recommended interval between 25% and 85%, which limits the autonomy of the vehicle to only 60%. They investigate the results of the widely used data generated by Schneider et al. (2014) by solving MILP to optimality. While such a strict approach makes a great effort to prevent SoC from contributing to battery deterioration, it highly shortens EVs' already limited driving distance. In cases where time windows of deliveries must be considered, this could greatly affect the feasibility region, resulting in more vehicles needed and larger overall driving distances.

The adverse effects of both high and low SoC on battery wear were also explored by Guo, Zhang, Huang and Huang (2022). The battery degradation costs were included in the Location Routing Problem (LRP) objective function to determine the optimal locations of recharging stations. The *Wear Cost* (WC) they used was inspired by the work of Han, Han and Aki (2014). To calculate these costs, some information on the batteries is required: the purchase cost of the battery, the cycle life and depth of discharge (DOD), and the achievable cycle count (ACC).

3 Problem description

EVRPTW-PR can be defined as follows: N customers with known demands are served by a homogeneous fleet of EVs with fixed load capacities. Each customer has a predetermined delivery time window and service duration, while the vehicles have limited driving ranges and, therefore, may need to recharge at one of the charging stations. The charging process takes a certain amount of time that depends on the desired increase of the battery level. All vehicles start and end their route at the depot, and partial recharging is allowed in this particular variant of the EVRPTW problem. Each EV departs from the depot with the battery fully charged, and if it gets charged at least once, it returns to the depot with a battery fully depleted. Otherwise, it may return with any battery level remaining.

The notation and formulation in this section follow the ones presented in Keskin and Çatay (2016). Let $V = U \cup R'$ be the set of all vertices, where $U \in \{1, ..., N\}$ represents the customers, and R' is the set of dummy vertices generated to permit several visits to each vertex in the set of R of recharging stations. The depot is denoted by both 0 and $N+1$. All routes must begin at 0 and end at $N+1$. For clarity of the constraints, we also introduce $R'_0 = R \cup \{0\}$ $V_0 = V \cup \{0\}$, $V_{N+1} = V \cup \{N+1\}$ and $V_{0,N+1} = V \cup \{0\} \cup \{N+1\}$. The set of arcs $A = \{(i,j) | i,j \in V_{0,N+1}, i \neq j\}$ represents the traveling between any two locations. Therefore, the problem can be summarized by a complete directed graph $G = (V_{0,n+1}, A)$.

Each electric vehicle has a fixed loading capacity C and battery capacity Q. The battery gets recharged at a charging rate of g. Furthermore, let d_{ij} and t_{ij} be the travel distance and travel time between locations $i, j \in V_{0,N+1}$ respectively. When a vehicle travels the arc $(i,j) \in A$, it consumes $h \cdot d_{ij}$ of its remaining battery, where h is the battery consumption rate.

Every customer $i \in U$ has a positive demand q_i ($q_i = 0 \forall i \notin U$), service time s_i ($s_i = 0 \forall i \notin U$)

and a time widow $[e_i, l_i]$. The service of a customer i cannot begin earlier than e_i or later than l_i, but it may end later.

There are several decision variables to consider. In order to track the vehicle routes, we introduce:

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \in A \text{ is traversed,} \\ 0 & \text{otherwise.} \end{cases}$$

Additionally, for every location $i \in V$, we define τ_i, the arrival time to the location, and u_i, the remaining cargo level upon arrival. Since partial recharging is allowed, we have two sets of decision variables to track the battery state of charge: y_i and Y_i, which track the battery state upon arrival to and departure from location i, respectively.

The problem can be defined as a mixed-integer program as follows:

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

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

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

$$\sum_{i \in V_0, i \neq j} x_{ij} - \sum_{i \in V_{N+1}, i \neq j} x_{ji} = 0 \qquad \forall j \in V \tag{4}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0(1 - x_{ij}) \leq \tau_j \qquad \forall i \in U \cup \{0\}, \forall j \in V_{N+1}, i \neq j \tag{5}$$

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

$$e_i \leq \tau_i \leq l_i \qquad \forall i \in V_{0,N+1} \tag{7}$$

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

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

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

$$0 \leq y_j \leq Y_i - (h \cdot d_{ij}) + Q(1 - x_{ij}) \qquad \forall i \in R'_0, \forall j \in V_{N+1}, i \neq j \tag{11}$$

$$y_i \leq Y_i \leq Q \qquad \forall i \in R'_0 \tag{12}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in V_0, \forall j \in V_{N+1}, i \neq j \tag{13}$$

The objective function 1 aims to minimize the total distance traveled. The connectivity of the customers and recharging stations are ensured by constraints 2 and 3, respectively. Constraints 4 are for flow conservation, while constraints 5 and 6 enforce the time feasibility between two visits when leaving from a customer location (or the depot) and a recharging station separately. The time window restrictions are represented by constraints 7. Additionally, constraints 5 - 7 work as a sub-tour elimination. Demand satisfaction of all customers is ensured by constraints 8 and 9. Furthermore, constraints 10 - 12 track the battery charge and stop it from dropping below 0. Finally, constraints 13 define decision variables as binary.

3.1 Battery degradation costs

When battery degradation is considered, the problem formulation remains almost identical, as stated above, but two new types of decision variables are introduced, and the objective function is expanded to include the additional costs that battery degradation contributes to. We define W_L and W_H as an average wear cost per unit energy charged below 25% and above 85%, respectively. The computation of these costs is inspired by Han et al. (2014). We introduce decision variables z_i^L and z_i^H, which represent the units of energy charged in location i in intervals 0 - 25% and 85 - 100%, respectively. These variables are directly affected by y_i and Y_i; therefore, two additional constraints are required to control them:

$$z_i^L = \min(0, 0.25Q - y_i) \qquad \forall i \in R' \cup \{N+1\} \quad (14)$$

$$z_i^H = \min(0, Y_i - 0.85Q) \qquad \forall i \in R'_0 \quad (15)$$

z_i^L is also computed for location $N+1$ because we want to track the SoC dropping below 25% while returning to the depot. Similarly, z_i^H is also defined for location 0 to indicate the SoC above 85% upon leaving the depot.

Then, the objective function becomes:

$$\min \sum_{i \in V_0, j \in V_{N+1}, i \neq j} d_{ij} x_{ij} + W_L \sum_{i \in R' \cup \{N+1\}} z_i^L + W_H \sum_{i \in R'_0} z_i^H \quad (16)$$

The first summation is the same as before, and accounts for the costs generated by the distance traveled. The second and third summations represent the costs of battery wear when the battery was below 25% and above 85% of its capacity, respectively. These costs are proportional to how much the recommended SoC range was violated.

4 Methodology

Keskin and Çatay (2016) propose an *Adaptive Large Neighborhood Search algorithm* to solve the EVRPTW-PR. The ALNS algorithm was introduced by Ropke and Pisinger (2006) and was designed to dynamically adjust and improve the search process by allowing the algorithm to visit infeasible solutions. In doing so, the algorithm is less likely than its predecessor LNS to be stuck in a local minimum. According to the authors, ALNS successfully improved the previously known best solutions for more than 50% of the examined problems. Therefore, ALNS has been widely applied in various routing problems. The following sections follow the methodology used by Keskin and Çatay (2016). Last section summarizes the adjustments for this approach that would keep the battery SoC at 25 - 85%, which we will refer to as recommended Soc.

4.1 Overview of ALNS approach

4.1.1 Construction of an initial solution

The pseudo-code of the algorithm for constructing an initial solution is presented in Algorithm 1. We start by initializing the current route by selecting the customer closest to the depot and adding them to the route. We proceed by iteratively attempting to add new customers to the current route until all customers are routed. We start every iteration by determining the insertion costs of each unrouted customer to every possible insert position in the current route. According to the exact insert position, arrival times for all customers can be calculated. Therefore, the insertion is considered possible if the route after insertion does not violate any of the routed customers' time windows and if the loading capacity can still satisfy the additional customer. If at least one of such insertions exists, we choose the customer insertion that results in the shortest total travel distance of the route. Otherwise, we initialize a new route by selecting an unrouted customer closest to the depot, adding them to this route, and proceeding with the next iteration. Suppose the battery at any point of the current route would drop below 0. In that case, we add a visit to the recharging station by performing a *Greedy Station Insertion*, described in Section 4.4.2.

Algorithm 1 Initial solution construction

```
Start a new route with the customer closest to the depot
while all customers are served do
    Calculate insertion costs of all unserved customers to the current route
    if no customer can be added then
        Start a new route with the unserved customer closest to the depot
    else
        Select the customer which increases the distance least and make the insertion
    end if
    if a recharging station is needed then
        Perform Greedy Station Insertion
    end if
end while
```

4.1.2 ALNS procedure

Algorithm 2 provides the pseudo-code for the ALNS procedure. The first step is constructing the initial solution described in Section 4.1.1. Then, the algorithm runs for a fixed number of iterations, destroying the current solution by removing individual customer or service nodes from a specific route or an entire route from the solution. Then, it repairs the solution by adding charging stations, removing customers from existing routes, or creating new ones. It does so by using four classes of algorithms: *Customer Removal* (CR), *Customer Insertion* (CI), *Station Removal* (SR), and *Station Insertion* (SI).

Each class of algorithms includes a few different approaches, and they are selected based on a probability that corresponds to their past performance in enhancing the solution. The performance of algorithm a is measured by adaptive weight w_a and a score π_a. Initially, all weights are equal, and all scores are set to 0. When an algorithm is used, its score is increased by σ_1, if the new best solution is found, σ_2, if a better solution than the previous is found, σ_3, if a worse solution than the previous is found, but it is accepted using the simulated annealing rule and 0 otherwise. Therefore, higher values of π_a indicate better performance. The performance weights are updated at the end of every segment s which lasts N_C iterations for customer, and N_R iterations for recharging station insertion/removal algorithm a according to the formula: $w_a^{s+1} = w_a^s(1-\rho) + \rho\pi_a/\theta_a$, where θ_a is the number of times it was used during segment s and ρ is the roulette wheel parameter. Then, the scores π_a are set to 0, and the probabilities of using each algorithm a in the next segment are determined by normalizing the weights of every algorithm class separately.

A worse than a previous solution can still contribute to the used algorithm's score if it is accepted according to the *Simulated Annealing* (SA) technique: if the solution uses fewer vehicles or the solution uses the same number of vehicles but results in a shorter total distance, then it is accepted. Additionally, suppose the number of vehicles is the same, but the distance is greater. In that case, the solution is accepted with a probability given by $e^{-f(X_{New}-f(X_{Current})/T}$, where X_{New} and $X_{Current}$ are the new and current best solutions respectively, $f(X)$ is the total distance of solution X and T is the current temperature, which is initially set to T_{init} and is reduced with every iteration by a factor of a cooling rate parameter $0 < \epsilon < 1$. T_{init} is set based on the initial temperature control parameter μ, ensuring that a solution which is μ% worse than the initial solution has a 50% chance of being accepted

Algorithm 2 ALNS algorithm

Generate an initial solution (Algorithm 1)
$j \leftarrow 1$
while Stop-criterion met **do**
  if $j \mod N_{SR} \equiv 0$ **then**
    Select SR algorithm and remove stations
    Select SI algorithm and repair solution
  else if $j \mod N_{RR} \equiv 0$ **then**
    for n_{RR} iterations **do**
      Select RRR or GRR algorithm and remove customers
      Select CI algorithm and repair solution
    end for
  else
    Select CR algorithm and remove customers
    if destroyed solution infeasible **then**
      Perform Greedy Station Insertion
      Select CI algorithm and repair solution
    end if
  end if
  Using SA criterion to accept/reject the solution
  $j \leftarrow j + 1$
  if $j \mod N_C \equiv 0$ **then**
    Update adaptive weights of CR and CI algorithms
  else if $j \mod N_S \equiv 0$ **then**
    Update adaptive weights of SR and SI algorithms
  end if
end while

4.2 Removal algorithms

4.2.1 Customer removal

For *Customer Removal* (CR), we use several algorithms as described by Keskin and Çatay (2016) and inspired by Demir, Bektaş and Laporte (2012) and Emeç, Çatay and Bozkaya (2016). The number of customers removed depends on the total number of customers n_c and is a random number $\gamma \sim \mathcal{U}(\underline{n_c}, \overline{n_c})$. The removed customers are then inserted into the removal list $\mathcal{L}$.

Random Removal: customers are selected randomly with equal probabilities to be chosen.

Worst-Distance and *Worst-Time Removal*: customers that result in higher costs have a higher probability of being chosen. For *Worst-Distance*, the cost is $d_{ji} + d_{ik}$, where i is the customer under consideration, j and k are their successor and predecessor, respectively. For *Worst-Time*, it is simply $|\tau_i - e_i|$. The customer with $\lfloor |\gamma| \lambda^{\mathcal{K}} \rfloor^{th}$ highest cost is chosen, where $\lambda \in [0, 1]$ is random and $\mathcal{K} \leq 1$ is the worst removal determinism factor.

Shaw Removal: this operation aims to remove a set of customers that are considered similar. It starts by randomly selecting customer i. Then, the relatedness measure for all remaining non-removed customers is calculated as $R_{ij} = \theta_1 d_{ij} + \theta_2 |e_i - e_j| + \theta_3 l_{ij} + \theta_4 |q_i - q_j|$. l_{ij} is a binary variable that takes the value -1 if customers i and j are in the same route and 1 otherwise, while θ_1 - θ_4 are the Shaw parameters. Then, all customers j are ranked according to R_{ij} in descending order, and $\lfloor |\gamma| \lambda^{\eta} \rfloor^{th}$ customer is chosen to be removed, where η stands for the Shaw removal determinism factor.

Proximity, *Time* and *Demand-Based Removal*: same as *Shaw Removal* but θ_1, θ_2, θ_4 is equal to 1 respectively while all the other Shaw parameters are set to 0.

Zone Removal: customers belonging to the same Cartesian plane zone get removed at once. First, corner points of the area are determined. Then, using the K-means clustering, customers are divided into n_z zones, a zone is randomly selected and all customers belonging to the zone are removed.

Random Route Removal (RRR): ω routes are chosen randomly, and all the customers in those routes get removed. The number of routes chosen ω is a random number between 10% and m_r% of the total number of routes.

Greedy Route Removal (GRR): similar to RRR, but instead of routes being chosen randomly, ω routes with the fewest customers are selected for removal.

Sometimes, removing a customer results in some charging of the EV no longer being necessary. To eliminate this inefficiency, we introduce two more algorithms: *Remove Customer with Preceding Station* (RCwPS) and *Remove Customer with Succeeding Station* (RCwSS), in which not only a customer from the removal list is removed, but also the station that is preceding or succeeding them is if such exists.

4.2.2 Station removal

For every N_{SR} iteration, station removal (SR) is performed, followed by station insertion operation. Similarly to γ in customer removal, σ stations are removed, and this number depends on the number of visits to the stations in the current solution. Again, several different algorithms are used to accomplish this:

Random Station and *Worst Distance Station Removal*: similar as *Random* and *Worst Distance Removal* for CR case.

Worst-Charge Usage Station Removal: stations visited with high battery levels get removed.

Full Charge Station Removal: stations where the vehicle is charged to its capacity are removed.

4.3 Route removal

In addition to these removal algorithms, the ALNS approach additionally aims to reduce the number of routes by performing RRR or GRR algorithm from Section 4.2.1 every N_{RR} iteration for n_{RR} times. These algorithms remove the entire route simultaneously, reducing the number of electric vehicles needed. After, the solution is repaired with CI algorithms (Section 4.4.1).

4.4 Insertion algorithms

4.4.1 Customer insertion

After some customers are removed, they must be inserted back into the solution to restore feasibility. The algorithms that do this are the following:

Greedy insertion: every removed node gets inserted into the cheapest possible position. We do this by computing all possible insertion costs for every customer $i \in \mathcal{L}$ as $d_{ji} + d_{ik} - djk$, where j is the preceding and k is the succeeding customer. The customer with the lowest possible insertion cost gets chosen first.

Regret-k Insertion: it works similar to *Greedy Insertion*, except for each customer $i \in \mathcal{L}$ we calculate the difference between the best and k^{th} best insertion costs and insert the customer which has the highest difference between the two first. Our application will use *Regret-2* and *Regret-3*.

Time-Based Insertion: same as *Greedy Insertion*, but the costs are now calculated according to the travel time: $t_{ji} + t_{ik} - tjk$.

Zone Insertion: same as *Time-Based Insertion*, but only a part of all routes in the solution are considered. Just like in *Zone Removal*, the solution is split into zones; one of them gets selected randomly, and only the routes within that zone may be expanded by the insertion.

After a customer insertion, the solution might become infeasible with respect to the battery state of charge. In such case, *Greedy Station Insertion* (See section 4.4.2) is performed.

As mentioned in 3, the vehicle starts the route fully charged, and if the recharging station is visited at least once, it returns to the depot with an empty battery. This rule also influences the recharge amount and, consequently, the battery state of charge. Let us summarize how different scenarios affect these values.

In case the vehicle is recharged only once, then there are two possible outcomes: (i) if a customer is added before the visit to the charging station, this insertion affects only the state of charge upon arrival at the station; (ii) if a customer is added after the visit to the station, the recharge amount is increased so that the EV returns to the depot with an empty battery. On the other hand, if the vehicle is recharged multiple times within the route and (iii) if a customer is added before the visit to the station, the procedure is the same as (i); (iv) if the customer is added after the visit to the first station, the recharge amount at the last station is increased so that the EV returns to the depot with an empty battery. However, if this violates any service time windows, a charge amount may be increased at a previous station instead.

4.4.2 Station insertion

When some charging stations are removed, the vehicle will likely no longer have enough battery charge to reach the next station or the depot. Therefore, we identify the location at which the battery charge drops below 0, and we restore feasibility by adding a visit to the station prior to the route. Remember that a station inserted may differ from the one previously removed, and each charging station may be visited multiple times. These three approaches may insert stations:

Greedy Station Insertion: similar to *Greedy Insertion*, but instead of checking all possible positions, the trip to the station is added right before arriving at a location where the battery would become negative. If this is unfeasible, insertion in earlier positions is attempted.

Greedy Station Insertion with Comparison: similar to *Greedy Station Insertion*, but this solution is additionally compared to the one in which we insert a station one position earlier and choose the cheaper option. If both options are infeasible, we apply *Greedy Station Insertion* for earlier positions.

Best Station Insertion: similar to *Greedy Station Insertion with Comparison*, but not just two positions prior to the visit to a location in which the battery would become negative are considered, but all positions since the depot or the last visit to the station.

SI approach is reiterated until the battery charge feasibility is restored for the entire solution. If this is impossible, the ALNS algorithm reverts to the last feasible solution and starts a new iteration.

4.5 Algorithm adjustments for avoiding battery degradation

The main difference when performing ALNS approach while considering attery degradation is that the vehicle SoC is restricted to stay between 25 - 85% of its full capacity. This mostly affects the frequency at which recharging stations should be visited, as well as the charging quantites that are determined in each of those visits. We summarize how different parts of ALNS algorithm are affected by this restriction:

Initial solution construction: The initial solution is constructed in the same manner as before, except that the visits to the source are scheduled before the battery drops to 25% of its capacity, instead of 0%.

Station Removal: when performing *Full Charge Station Removal*, stations, where vehicles are charged to the maximum possible amount, that is 85% of total capacity, get removed first.

Customer Insertion: the insertion operations are performed identically as before, but the battery charge feasibility differs. First of all, the vehicles are no longer assumed to leave the depot fully charged. Instead, they are charged to 85% of their capacity. Furthermore, if the vehicle is charged at least once during the route, they return to the depot with 25% of battery life remaining. These differences are reflected in charging quantites.

Station Insertion: just like in construction of initial solution, station insertion is considered as soon as the vehicle's battery drops to 25%. This gets reflected in all SI operations.

5 Computational study

In this section, we will present the results of solving EVRPTW-PR problem in two ways: by solving MILP with GUROBI and by applying the ALNS approach. First, the battery degradation will be ignored while obtaining the solution. Then, we will solve the same problem by including the battery degradation costs to MILP formulation and by adjusting the ALNS approach in a way that maintains the battery at the recommended level between 25 - 85%. For all solutions, the costs will be calculated to both include and disclude the battery degradation costs as they are described in 3.1 so we could make a better comparison of the results.

The data used in this paper are the widely used instances generated by Schneider et al. (2014) which were obtained through Dominik (2019). To avoid complexity, we will not be tuning the parameter values ourselves. Instead, we will use the values obtained by Keskin and Çatay (2016).

6 Conclusion

References

BU. (2023, Oct). *Bu-808: How to prolong lithium-based batteries.* Retrieved from `https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries`

Cataldo-Díaz, C., Linfati, R. & Escobar, J. W. (2022, Jan). Mathematical model for the electric vehicle routing problem considering the state of charge of the batteries. *Sustainability*, *14*(3), 1645. doi: 10.3390/su14031645

Conrad, R. G. & Figliozzi, M. A. (2011). The recharging vehicle routing problem. *Proceedings of the 2011 Industrial Engineering Research Conference.*.

Dantzig, G. B. & Ramser, J. H. (1959, Oct). The truck dispatching problem. *Management Science*, *6*(1), 80–91. doi: 10.1287/mnsc.6.1.80

Demir, E., Bektaş, T. & Laporte, G. (2012, Dec). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346–359. doi: 10.1016/j.ejor.2012.06.044

Dominik, G. (2019). *The electric vehicle-routing problem with time windows and recharging stations.* Retrieved from `https://data.mendeley.com/datasets/h3mrm5dhxw/1`

EEA. (2024, Apr). *Eea greenhouse gases - data viewer.* Retrieved from `https://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer`

Emeç, U., Çatay, B. & Bozkaya, B. (2016, May). An adaptive large neighborhood search for an e-grocery delivery routing problem. *Computers amp; Operations Research*, *69*, 109–125. doi: 10.1016/j.cor.2015.11.008

Felipe, , Ortuño, M. T., Righini, G. & Tirado, G. (2014, Nov). 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. doi: 10.1016/j.tre.2014.09.003

Guo, F., Zhang, J., Huang, Z. & Huang, W. (2022, Jul). Simultaneous charging station location-routing problem for electric vehicles: Effect of nonlinear partial charging and battery degradation. *Energy*, *250*, 123724. doi: 10.1016/j.energy.2022.123724

Han, S., Han, S. & Aki, H. (2014, Jan). A practical battery wear model for electric vehicle charging applications. *Applied Energy*, *113*, 1100–1108. doi: 10.1016/j.apenergy.2013.08.062

Irle, R. (2024, May). *The electric vehicle world sales database.* Retrieved from `https://ev-volumes.com/`

Keskin, M., Laporte, G. & Çatay, B. (2019, Jul). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers amp; Operations Research*, *107*, 77–94. doi: 10.1016/j.cor.2019.02.014

Keskin, M. & Çatay, B. (2016, Apr). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, *65*, 111–127. doi: 10.1016/j.trc.2016.01.013

König, A., Nicoletti, L., Schröder, D., Wolff, S., Waclaw, A. & Lienkamp, M. (2021, Feb). An overview of parameter and cost for battery electric vehicles. *World Electric Vehicle Journal*, *12*(1), 21. doi: 10.3390/wevj12010021

Pasha, J., Li, B., Elmi, Z., Fathollahi-Fard, A. M., Lau, Y.-y., Roshani, A., ... Dulebenets, M. A. (2024, Mar). Electric vehicle scheduling: State of the art, critical challenges, and future research opportunities. *Journal of Industrial Information Integration*, *38*, 100561. doi: 10.1016/j.jii.2024.100561

Rastani, S. & Çatay, B. (2021, Nov). A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows. *Annals of Operations Research*, *324*(1–2), 761–793. doi: 10.1007/s10479-021-04320-9

Ropke, S. & Pisinger, D. (2006, Nov). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472. doi: 10.1287/trsc.1050.0135

Schneider, M., Stenger, A. & Goeke, D. (2014, Nov). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*(4), 500–520. doi: 10.1287/trsc.2013.0490

Solomon, M. M. (1987, Apr). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265. doi: 10.1287/opre.35.2.254

Wang, H. & Cheu, R. L. (2013, Jan). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. *Transportation Research Record: Journal of the Transportation Research Board*, *2352*(1), 1–10. doi: 10.3141/2352-01

A Programming code