Integrating electric vehicles in a the Two-Echelon Vehicle Routing Problem

Econometrics and Operations Research

FEB63007 || *Bachelor Thesis*

Tess Zijlstra
483750

Galindo Pecin, D.
Supervisor

Spliet, R.
Second Assessor

July 4, 2021

Abstract

When electric vehicles execute routes in a Two-Echelon distribution system, route planning has to deal with the limited driving range of the electric vehicles. As a consequence, visits to charging stations must be integrated into the route planning. This paper elaborates on an existing method to solve the Electric Two-Echelon Vehicle Routing Problem (E2E-VRP). This study extends this method by incorporating a non-linear charging function and examines the effect on the total cost of partial recharging. To obtain solutions for the E2E-VRP, the metaheuristic LNS-E2E, proposed by Breunig et al. (2019) was modified to LNS-E2E*. This heuristic uses a Large Neighbourhood Search (LNS) algorithm to achieve near-optimal solutions with a minimal total cost. The total cost of the tours includes distance costs and usage costs. The performance of the solutions obtained by LNS-E2E* diminished when the problem size increased. To examine the effect of partial recharging when a non-linear charging function is assumed, a method is proposed to extend LNS-E2E* such that also charging costs are included. The obtained results imply that the total cost reduces and the number of visits to charging points increases when partial recharging is allowed and, a non-linear charging function is assumed.

Contents

1 Introduction **1**

2 Theoretical background **2**

3 Relevant literature **3**

4 Problem description **5**

4.1 E2E-VRP 5

4.2 Integrating charging time 6

5 Research methodology **7**

5.1 Solving the E2E-VRP 7

5.1.1 Large Neighborhood Search 8

5.2 Integrating charging function and time 12

5.2.1 Non-linear charging function 12

5.2.2 Translating charging time into charging costs 13

5.2.3 Full and partial charging 13

6 Computational study **14**

6.1 Data 15

6.2 Parameter settings 15

6.3 Numerical Results 15

6.3.1 LNS-E2E* 15

6.3.2 Full and partial charging 18

7 Conclusions and future research **19**

8 Discussion **20**

1 Introduction

Due to the rising demand for product deliveries combined with urban sustainable development goals, sustainable transport networks became increasingly important in the field of city logistics. Because transportation by combustion engine vehicles causes water, air, and noise pollution, the question of how to integrate sustainable vehicles in transportation networks became an important research topic. The Vehicle Routing Problem (VRP) plays a central role in the study of transport networks. The VRP is a class of problems in which finding efficient routes to visit customers using a fleet of capacitated trucks, starting and finishing at a central depot, is the purpose. Finding efficient routes is crucial for both companies and clients, as efficient routing reduces travel time and operational costs for companies and therefore reduces selling prices for customers. The VRP has many variants to capture different characteristics from the real world, such as VRPs with time windows or with multiple depots. In this study, another variant of the VRP is further examined; the Electric 2-Echelon Vehicle Routing Problem (E2E-VRP). This variant is about integrating electric vehicles (EVs) in a two-echelon transport network, where the transport network of one depot and a set of customers is extended by including intermediate distribution centers.

This study elaborates on the work of Breunig et al. (2019), who introduced the E2E-VRP. In the classic 2-Echelon Vehicle Routing Problem (2E-VRP), decisions, when and where to refuel the vehicles, are neglected, because petrol pumps are far more universal than charging stations (CS). In addition, the driving range of combustion engine vehicles is much larger than the range of EVs. So, deciding when and where to charge during the execution of a route becomes crucial. Breunig et al. (2019) were the first to integrate EVs in a two-echelon transport network. The proposed model minimizes only the cost for distance and the usage of vehicles while neglecting possible charging costs. As the charging cost depends on the recharge method, battery level, charging function, recharge amount, and availability of charging stations, it is necessary to take these elements into account when determining efficient routing for practical use. Therefore, this study will focus on extending the model from Breunig et al. (2019) such that the time consumed by visiting recharging stations is integrated. This research aims to determine the effect of incorporating charging costs while determining efficient routes instead of considering only the distance and usage costs. The integration of the charging costs depends on the assumption of a non-linear charging function. That means the charging speed decreases after a specified battery level, and it could be very time-consuming and therefore inefficient to fully charge the vehicles if this is unnecessary. Because time is a critical factor in transportation, it is valuable to take the time needed to recharge into account. This study will be the first that extends the E2E-VRP to a more practical problem in terms of charging costs and creates a more realistic framework to apply the solving method in practice. The central research question is as follows:

"How will including charging costs affect the total cost of the E2E-VRP when partial charging is allowed and charging functions are assumed to be non-linear?"

To answer this question, the research has two phases. In the first phase, the metaheuristic LNS-E2E*, proposed by Breunig et al. (2019), is replicated to be able to solve the E2E-VRP without integrating the charging costs. The second phase, referred to as the extensive part, extends the LNS-E2E* such that the method includes charging costs during the determination of efficient routes.

This paper is structured as follows. Section 2 presents some theoretical background about the VRP, 2E-VRP, and E2E-VRP. In Section 3, a summary of relevant literature related to the variants of the electric VRP to outline the existing solution approaches. Second 4 formulates the problem mathematically, and in Section 5 the research methodology is expounded. Section 6 presents and discusses the results and, Section 7 presents the conclusions and recommendations.

2 Theoretical background

This paragraph elaborates on the short explanation of the VRP mentioned in the introduction. In the general VRP, 'customers' have a certain type of demand that must be satisfied. Customers are served by a fleet of vehicles, starting and finishing their routes at a central depot. The VRP, introduced by Dantzig & Ramser (1959) as the "Truck Dispatching Problem", is a generalization of the well-known Traveling Salesman Problem (TSP). As the TSP is known to be NP-hard, the VRP is NP-hard as well. This means that computational time to find an exact solution grows exponentially when the problem size increases. Therefore, heuristic approaches are used to solve problems when exact algorithms take too much computational effort. Heuristic procedures discover solutions and try to find one that is convenient. Many heuristic approaches were proposed in former literature to create solutions for several types of VRPs.

Because the E2E-VRP is a special case of the 2-Echelon VRP (2E-VRP), the 2E-VRP is explained first. Gonzalez-feliu et al. (2008) introduced the 2E-VRP as the simplest type of the larger category: Multi-Echelon-VRP. In Multi-Echelon supply chains multiple VRPs which occur at different levels are solved at the same time. The 2E-VRP considers two levels: the first level has a *central depot* and intermediate distribution centers, called *satellites*. Trucks transfer a type of commodity from the depot to the satellites, which make the *first level tours*. The second level consists of the satellites and customers. Smaller vehicles, for example, vans, are located at these satellites and deliver the desired commodity to

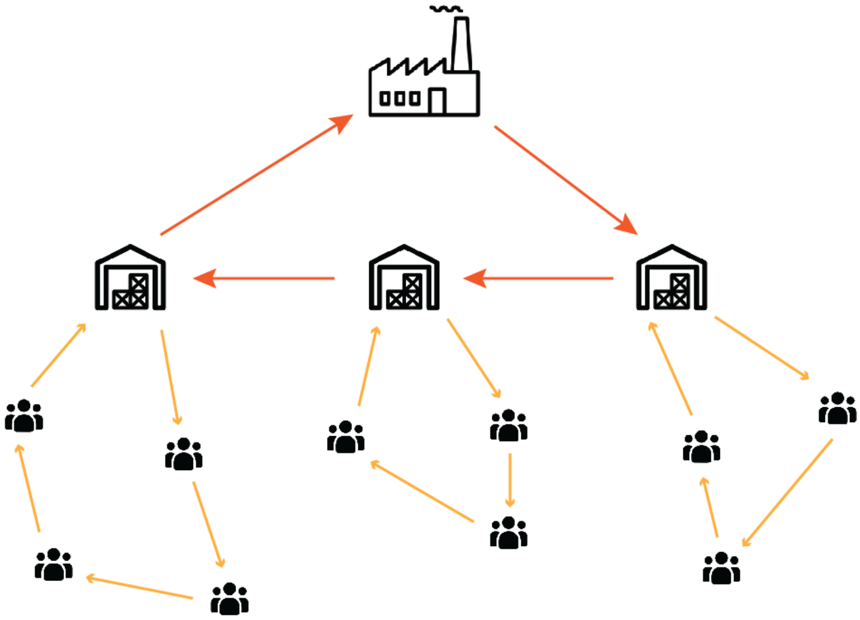


Figure 1: A 2E-VRP transport network, with first level tours (orange tour) and second level tour (yellow tours)

the customers, which create the *second level tours*. Figure 1 illustrates a simple example of the 2E-VRP. The integration of electric vehicles in this network makes the electric 2E-VRP, E2E-VRP. That means that charging stations (CS) are also part of the second-level network.

3 Relevant literature

Many solutions approach for variants of the electric VRP class can be found in the literature. These variants capture different characteristics of real-world transport networks. Researchers try to find solution approaches that create convenient results in acceptable computational time. This section summarizes the relevant literature for this study.

When introducing the 2E-VRP, Gonzalez-feliu et al. (2008) proposed a Mixed-Integer Programming (MIP) model formulation to solve the problem. Their approach performs well for small problem instances, but the method did not succeed in finding satisfying results for larger problems. Hemmelmayr et al. (2012) proposed a new solution method for the 2E-VRP and model the classic Location Routing Problem (LRP) as a 2E-VRP. They apply the heuristic approach called Adaptive Large Neighborhood Search (ALNS) on both problems. They found an approach for transforming LRP instances into 2E-VRP instances. As a result, the same ALNS algorithm works for the LRP successfully too. Their method outperforms former approaches to solve the VRP for large-sized problems.

As the second-level routes are executed by electric vehicles, the E2E-VRP is also part of the Green-VRP (G-VRP) category. The G-VRP was first formulated by Miller-hooks & Erdogan (2012), including a class of VRPs with limited fuel capacity and limited fuel stations available. They propose a solution approach where two classic heuristics are customized to the G-VRP. Solution approaches of the Electric VRP (EVRP) were also proposed in other studies such as Lin et al. (2016). Breunig et al. (2019) introduced the E2E-VRP, which integrates electric vehicles in the second level of this multi-tier transportation network. This study is based on their research and data.

Felipe et al. (2014) introduced the EVRP-MTPR (multiple technologies & partial recharging). They distinguish three charging types, with each a different cost and speed. To solve their problem, they proposed an exact algorithm, a local search procedure, and an algorithm based on Simulated Annealing (SA). They concluded that for large problem instances SA performed best. Moreover, they concluded that allowing partial recharging and having multiple charging technologies provided significant cost and energy savings.

In 2016, Keskin & Çatay studied the EVRPTW-PR (time windows & partial recharging). They propose an ALNS heuristic to create feasible solutions for the problem. After creating an initial feasible solution, they apply removal and reinsertion operators for customers and charging stations. The algorithm also searches for more efficient locations of charging stations and sees if partially charging results in better solutions. They concluded that routes were improved significantly when partial recharging is allowed.

Montoya et al. (2017) introduced the EVRP for nonlinear charging functions. In general, charging functions are non-linear as the voltage changes during the charging process. The aim is to find an optimal recharging policy by adaptively deciding the route. As a linear approximation of the charging function could result in infeasible or impractical solutions, it is useful to integrate more accurate charging functions. To solve the EVRP-NL, their proposed metaheuristic combines iterated local search (ILS) and a heuristic concentration (HC). In their study, they distinguish three charging types, similar to Felipe et al. (2014), with each type having its own non-linear charging function. They concluded that allowing partial recharging, using en-route charging, and integrating non-linear charging functions resulted in good solutions.

Sweda et al. (2017) studied the EVRP with adaptive routing and recharging for range-constrained vehicles. They consider a network where on each node the possibility to recharge exists (nodes without charging station have a recharge probability of zero). The recharge stations could be unavailable, whereby the driver should wait until it becomes available. For each node, the availability probability and expected waiting time are known. Every time the driver stops to recharge, stopping, recharging, and overcharging costs are considered. They propose two heuristics to obtain solutions for their model.

The E-VRP with Non-Linear charging and Load-Dependent discharging and Capacitated Charging Stations (E-VRP-NL-LD-CCS) was introduced by Kancharla & Ramadurai (2020). They apply an ALNS algorithm to create solutions for this complicated problem. Their solution approach was tested by using benchmark instances from Montoya et al. (2017) and they obtained better results by using ALNS instead of the metaheuristic ILS+HC.

4 Problem description

In this section, the particular E2E-VRP of this study is explained and formulated in more detail. First, the replication part is formulated properly, where after integrating charging time is clarified.

4.1 E2E-VRP

The E2E-VRP of this study contains one central depot, equipped with a homogeneous fleet of m^1 combustion engine vehicles, referred to as trucks, with cargo capacity Q_1. Trucks execute the first level tours, which start from the depot, deliver the desired commodity at the satellites and return to the depot. Only one type of commodity is considered. The satellites are equipped with several identical electric vehicles (EVs) where the number of EVs could be different per satellite. The EVs have a battery capacity L, which is never exceeded. The maximum number of EVs in use is m^2 and cannot exceed the total number of EVs located at the satellites. EVs have a cargo capacity of $Q_2 < Q_1$ to deliver the desired amount of commodity to the customer. Tours performed by EVs are the second level tours, which start at a satellite, visit several customers and possible charging stations and return to the same satellite.

To model the problem mathematically, the E2E-VRP is formulated as a multigraph. Because the first and second level are a different VRP and are solved after each other, they are formulated apart as $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and will together form the multigraph. Considering the sets of satellites; N_S, charging stations; N_R, customers; N_C and the depot; $\{0\}$, the graphs are formulated as follows, first level and second level respectively:

$G_1 = (V_1, E_1)$, where $V_1 = \{0\} \cup \{N_S\}$ and $E_1 = \{(0, l) : l \in N_S\} \cup \{(k, l) : k, l \in N_S, k < l\}$

$G_2 = (V_2, E_2)$, where $V_2 = \{N_S\} \cup \{N_C\}$ and $E_2 = \{(i, j) : i, j \in N_S \cup N_C, i \neq j\} \backslash \{(i, j) : i, j \in N_S\}$

Every edge $e = (i, j), e \in E_1 \cup E_2$ has a cost equal to the Euclidean distance between i and j. In G_2, every edge is associated with possible charging station visits, called CS-detours, illustrated in Figure 2. 'Possible' means that the consumption between i and the CS, and between the CS and j will not exceed the maximum battery capacity of the electric vehicles. Every CS-detour has a cost and consumption. The cost equals the total Euclidean distance of the detour and the consumption equals just the consumption needed between CS and j, as the EV is charged at the CS. Additional to the formulation, the following assumptions are made:

1. Charging stations can be used several times, but consecutive visits to charging stations are not allowed;
2. Second-level edges with a consumption higher than the maximum battery capacity of an EV are excluded from G_2;
3. Each first level tour starts and ends at the depot, each second level tour ends at the satellite where the tour started;
4. Each satellite also hosts a charging station;
5. Split deliveries to customers or satellites are not allowed;
6. All EVs are fully charged when a tour is started;
7. For solving the E2E-VRP in replication, EVs are fully charged at charging stations.

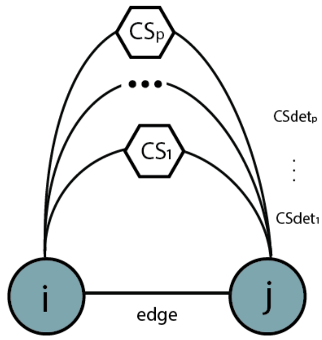
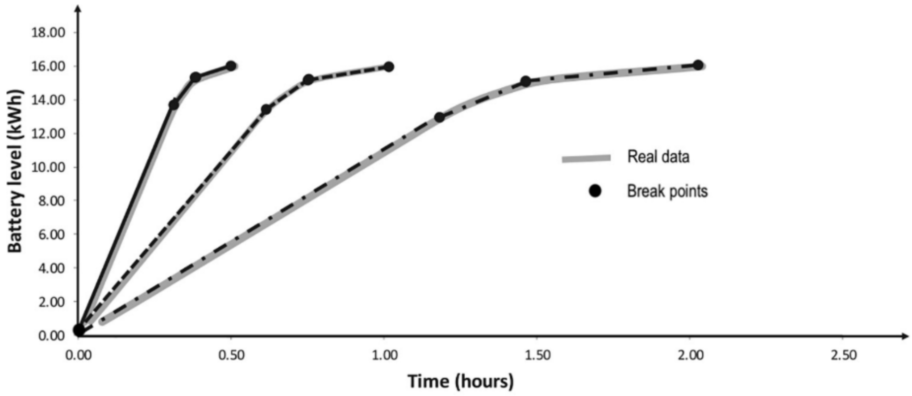


Figure 2: Edge (i,j) with corresponding CS-detours

4.2 Integrating charging time

In the extensive part, efficient routes are determined by considering distance cost, usage cost, and charging cost. These charging costs rely on the time spent at a CS. As mentioned before, charging time depends on several elements. In this study, the following elements are considered:

1. The *charging function* is assumed to be non-linear. It is a concave function, where the charging speed diminishes when the battery reaches a higher battery level. The charging function is adopted from Montoya et al. (2017), in which they use a battery with a capacity of 16 kWh. Figure 7 shows the charging function for three charging types. The authors use an average consumption rate of 125 Wh/km, which corresponds to a distance of 128 km. In Breunig et al. (2019) a battery capacity of 1000 corresponds to a distance of 100 km. Therefore, a battery of 16 kWh in Montoya et al. (2017) is treated similarly to a battery with a capacity of 1280 units in Breunig et al. (2019). In their sensitivity analysis, Breunig et al. (2019) use data sets with batteries of the following capacities: 800, 900, 1000, ... , 1700, with twenty different instances for each capacity. To be sure the non-linear charging functions are most accurate according to former literature, the instances for a battery with a capacity of 1300, closest to 1280, are adopted. This study considers only the moderate charging type, with a charging power of 22 kWh/h.



Type of CSs	Slow	Moderate	Fast
Charging power (kWh/h)	11	22	44
Piecewise linear approximation	— · —	------	——

Figure 3: Non-linear charging functions of three charging types for a 16 kWh battery, adopted from Montoya et al. (2017)

2. The *recharge amount* is an important feature to determine because partially recharging is allowed. Therefore, the assumption from Keskin & Çatay (2016) is adopted in which they determine the desired amount during the ALNS process based on the fact that an EV should return to the depot with an empty battery.

5 Research methodology

To be able to make reliable comparisons between the results obtained by Breunig et al. (2019), the first part of the methodology is about replication of their metaheuristic. The second part consists of the approaches needed to formulate an answer to the research question.

5.1 Solving the E2E-VRP

Breunig et al. (2019) apply a metaheuristic, called LNS-E2E, to obtain their solutions for problems with medium and large instances. Their metaheuristic combines the well-known Large Neighborhood Search (LNS) algorithm and a Local Search (LS) procedure to improve the solution obtained from LNS. This study implements a modification of LNS-E2E: LNS-E2E*. Through the limited time of this research, LNS-E2E* is less advanced than LNS-E2E. LNS-E2E* disregards the local search algorithms. Algorithms 1 and 2 on the next page show the pseudocode of both algorithms.

Algorithm 1: LNS-E2E

$S^{best} \leftarrow \varnothing$
while *the stopping criterion is not met* **do**
  $S \leftarrow LocalSearch(Repair(\varnothing))$
  for $i \leftarrow 0$ ***to*** i_{max} **do**
    $S^{cur} \leftarrow LocalSearch(Repair(Destroy(S)))$
    if $Cost(S^{cur}) < Cost(S)$ **then**
      $S \leftarrow S^{cur}$
      $i \leftarrow 0$
    end
  end
  if $Cost(S) < Cost(S^{best})$ **then**
    $S^{best} \leftarrow$ S
  end
end
return S^{best}

Algorithm 2: LNS-E2E*

$S^{best} \leftarrow \varnothing$
while *the stopping criterion is not met* **do**
  $S \leftarrow Repair(\varnothing)$
  for $i \leftarrow 0$ ***to*** i_{max} **do**
    $S^{cur} \leftarrow Repair(Destroy(S))$
    if $Cost(S^{cur}) < Cost(S)$ **then**
      $S \leftarrow S^{cur}$
      $i \leftarrow 0$
    end
  end
  if $Cost(S) < Cost(S^{best})$ **then**
    $S^{best} \leftarrow$ S
  end
end
return S^{best}

5.1.1 Large Neighborhood Search

The Large Neighborhood Search algorithm tries to find better solutions by destroying and repairing an initial solution many times. The algorithm starts by selecting an initial solution, chosen as the first current solution. The current solution is a set of tours that satisfy all customers, sometimes referred to as 'tour schedule'. By executing destroy and repair operators a new solution is created. If the newly created solution performs better than the current solution, the new solution becomes the current solution, and the process of destroying and repairing starts again. When the new solution performs worse, the current solution stays the same and the process continues. Appendix B contains a description of the implemented code of LNS-E2E*.

Destruction phase

In the destruction phase, one of the following three destroy operators is chosen, all could be chosen with the same probability:

1. *Related nodes removal:* this operator selects first a random 'seed' customer to remove. Secondly, a random number of its closest neighbors is removed from the existing tours. The total number of customers to remove is $p_1 n_c$, where n_c indicates the total number of customers.
2. *Random routes removal:* this operator removes one or more tours from the current solution. The customers from these tours are added to the set of removed customers. The number of tours to remove is chosen randomly in the interval $[0, p_2 L]$, where L represents the minimum number of tours needed to satisfy the total customers' demand.

3. *Close satellite:* A random satellite closes if the remaining satellites maintain a feasible solution in terms of capacity. The related customers are removed from the solution and the satellite stays closed in the phase of reconstruction.

After executing one of the destroy operators, the following operators could be chosen with associated probabilities p_4 and p_5:

4. *Open all satellites:* In which possibly closed satellites are opened for the reconstruction phase.
5. *Remove single customer routes:* This operator searches for single customer tours and removes these customers if found.

Reconstruction phase

The first phase results in a destroyed solution and a list of customers and possible satellites to reinsert and re-open. The second phase of LNS is about reconstructing the destroyed solution. This goes by inserting customers in the destroyed solution first, such that cargo capacity constraints are satisfied and battery level constraints are neglected. Secondly, the algorithm creates completely new first-level tours, because the commodity demand per satellite might have changed a lot in the first step. In the last step, tours become feasible in terms of battery consumption. To achieve this, a dynamic programming (DP) method searches for least-cost tours that now satisfy all constraints. If it happens that no feasible solution can be found, a penalty in terms of costs is assigned to the solution. In this way, the solution becomes very 'expensive' and it will not be chosen as best solution. The following paragraphs explain the described steps in detail.

Step 1: Inserting customers

The first step reinserts the removed customers by applying a simplified version of the cheapest insertion heuristic (CIH). Customers are chosen randomly from the set of removed customers and reinserted on its cheapest location. That means that the extra cost of reinserting the customer in the existing tour schedule is minimized. For every possible inserting location in each tour, the cost for inserting the customer at this place is determined and, the cheapest location is chosen. When customers are inserted by the CIH, the capacity constraints are satisfied. This means, among others, that every inserted customer receives their demand fully and is only visited in one tour, such that there are no split deliveries. Also, the maximum cargo capacity of electric vehicles is never exceeded. Because the ordering of inserting the customers is random, tours might become infeasible, which is possible when customers with high demand are inserted late. If this happens, the algorithm starts again but will insert the customers in the order of non-increasing demand.

Step 2: Creating new first level tours

From the resulting second-level tour schedule, the total demand for each satellite is known, so, the first-level tours can be determined. Beforehand, the algorithm determines if the demand for a satellite is higher than the capacity of one truck. If so, trucks visit the satellite until its demand is smaller than the truck capacity. After this step, the simplified CIH explained in step 1 is used to create the first-level tours.

Step 3: Inserting CS by using a DP algorithm

In the third step, second-level tours must become feasible in terms of battery consumption. The DP algorithm searches for the cheapest possible tour while constantly satisfying the battery capacity constraints. The DP algorithm receives a route as input, which starts at a satellite, visits a fixed order of customers, and returns to the same satellite. The DP algorithm then finds the best feasible placement of CS visits for this route. Figure 4a represents a simple example of such a route, where three customers are visited. Because every edge has a set of CS-detours, and customers are visited in a particular order, a directed graph can be formulated in which the satellite and customers are the vertices and all possible routes between consecutive nodes are the edges. Figure 4b illustrates an example of such a graph corresponding to the sequence from Figure 4a.

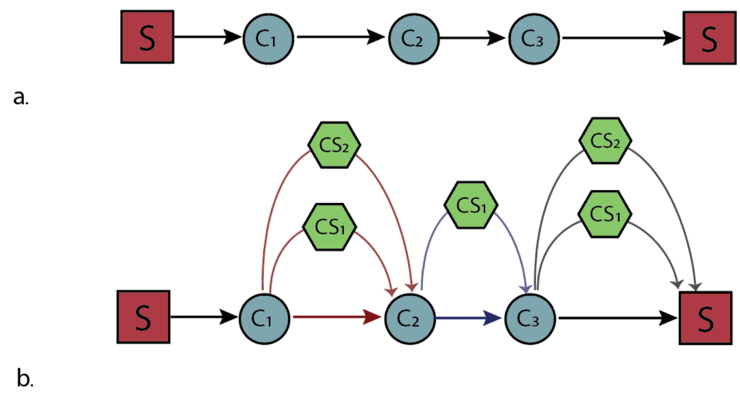


Figure 4:
a: The sequence of vertices for a tour
b: The directed graph associated to the sequence

Because different tours are possible, the algorithm should keep track of the cheapest tour found. To find the cheapest tour, keeping track of three elements is important: 1) the cost, 2) consumption and therefore battery level (feasibility) and, 3) the chosen edges. The algorithm explores every partial tour from the satellite to the next customer that satisfies the battery constraints. So first S $\rightarrow$ C1 then S $\rightarrow$ C1 $\rightarrow$ C2, and so on, and only keeps non-dominated partial tours. That means, given two partial tours, namely P1 and P2, if P1 has a lower total cost and less or equal battery consumption than P2, then P2 can be discarded. Otherwise, without this dominance rule, the DP algorithm may fail as the number of possible partial tours can grow exponentially. Figure 5 illustrates this process. Looking at the edges between the first and second customer, we have three possibilities.

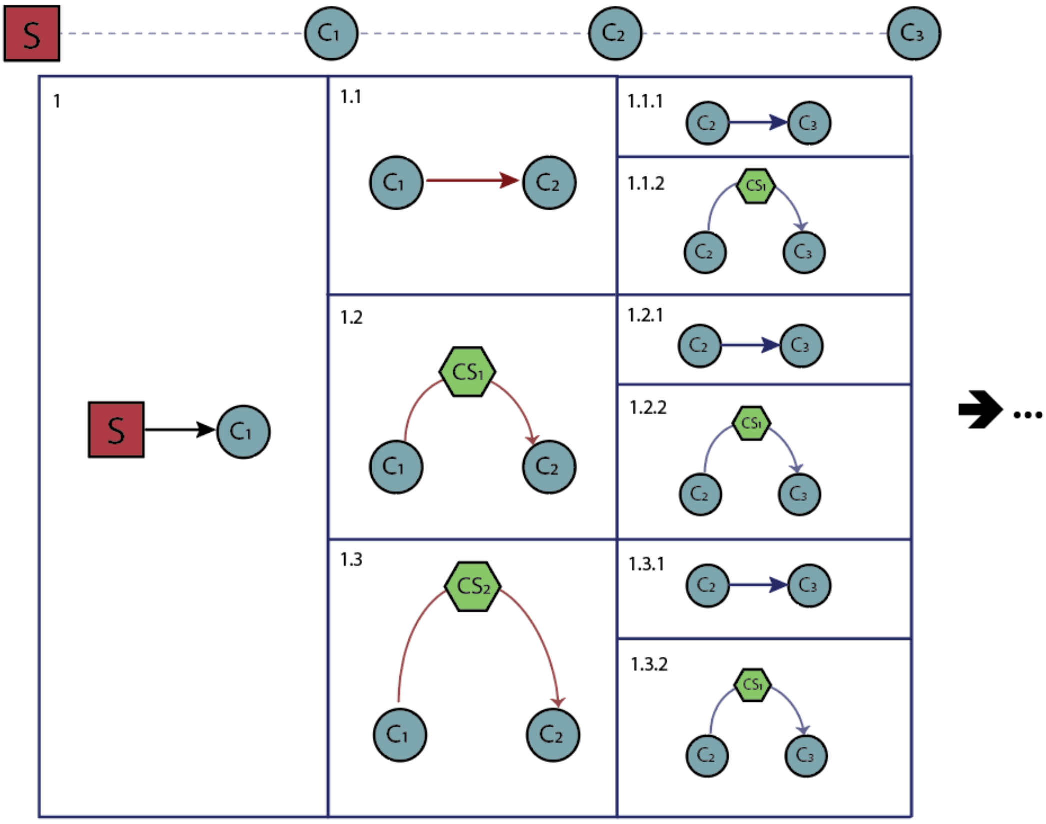


Figure 5: Process of finding all possible routes

The algorithm determines for each possibility the cost and consumption and saves the tour if it is feasible and non-dominated, thus, satisfies the battery capacity constraints. From customer 2 to 3, two routes are possible. We add these possibilities to the tours that were saved (second column) and determine the cost and consumption for every possible tour. This process continues until the satellite is reached again. Then, the cost of all tours that were kept at the last step by the DP algorithm are compared and the cheapest tour is chosen and composed by backtracking this tour.

Updating the solution

After LNS, the resulting (new) solution is compared with the initial (current) solution and updated if the new solution performs better. This process continues until a stopping criterion is satisfied. While executing LNS-E2E*, the algorithm keeps track of the best feasible solution and returns this solution when a maximum number of iterations is reached.

5.2 Integrating charging function and time

To integrate the charging function and the charging time, several aspects are of importance. Firstly, how to adapt the charging function from Montoya et al. (2017) matters, as the authors use different data than Breunig et al. (2019). Moreover, for both fully charging and partial charging, the recharge time should be added to the cost considered by Breunig et al. (2019), which includes distance and usage costs. To achieve this, charging time is translated to units that coincide with the distance and usage units. The following paragraphs explain these aspects in more detail. Important to note is that there is no difference in cost for travel time and charging time.

5.2.1 Non-linear charging function

To adopt the charging function from Montoya et al. (2017), illustrated in Figure 7, the numbers along the axes are converted to fit the data used in this study. As mentioned in the problem description, a battery of L = 1300 is considered as this is most comparable to a 16 kWh battery. To incorporate the non-linear

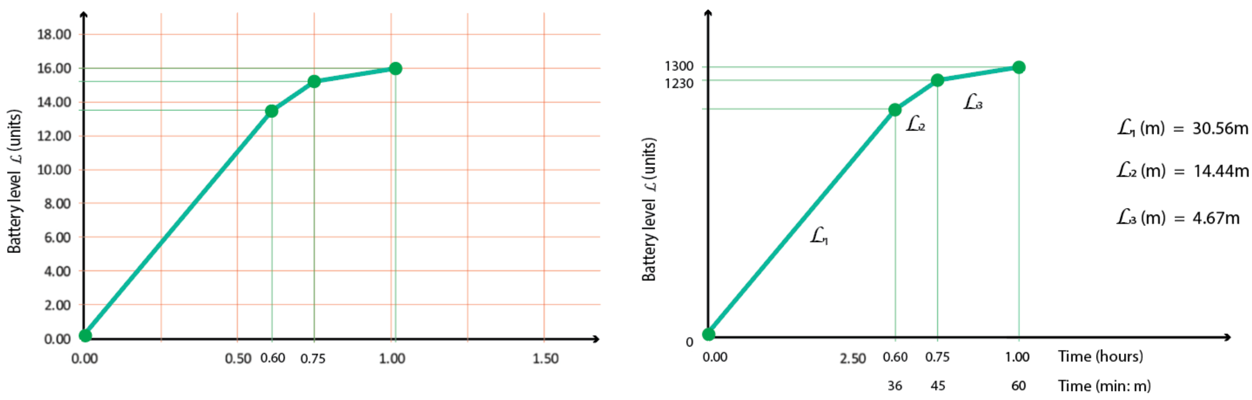


Figure 6: Piecewise linear approximation for the charging function of a 16 kWh battery (left) and a battery of L = 1300 (right)

behavior of the charging function, the function is piece-wise linearly approximated. Three different linear functions L_1, L_2 and L_3 construct the charging function, shown in Figure 6. These functions indicate that the charging speed when charging to a battery level of L = 1100 is approximately 30.56 battery units per minute. Charging between L = 1100 - 1230 continues at a rate of 14.44 battery units per minute, and from L = 1230 - 1300, the speed is diminished to 4.67 battery units per minute. After slightly more than one hour, the battery is charged fully. Because the function is an approximation, this number is rounded to one hour for convenience.

5.2.2 Translating charging time into charging costs

Breunig et al. (2019) consider distance units that correspond to 0.1 km. Usage costs are in the same units as distance costs, so translating time into distance units enables us to add time to the total cost. Felipe et al. (2014) use the same energy consumption rate for an EV as Montoya et al. (2017), and they assume an average speed of 25 km/h, that is 250 distance units for 60 minutes. Adopting this average speed means that each minute travel time when executing a route adds approximately 4.17 to the route its cost. Likewise, each minute of charging must add 4.17 to the total cost, as travel time and charging time are treated similarly.

5.2.3 Full and partial charging

To determine how partial charging affects the cost of fully charging, a benchmark is created first. With the assumption of deterministic consumption that is known in advance for each edge, the consumption of each segment between charging stations becomes available. With this information, it becomes possible to determine the battery level of the EV when it arrives at a CS in the tour. So, when we have this battery level and the charging function, we can calculate the time needed to charge the EV fully. And, in turn, as we know the consumption needed for each segment of the route, we also know the necessary battery level for partial charging. Assuming it is cheapest to avoid remaining battery capacity when arriving at a CS or when finishing a route, the EVs are charged partially to the exact level needed to perform the next segment. Figure 7 illustrates this process.

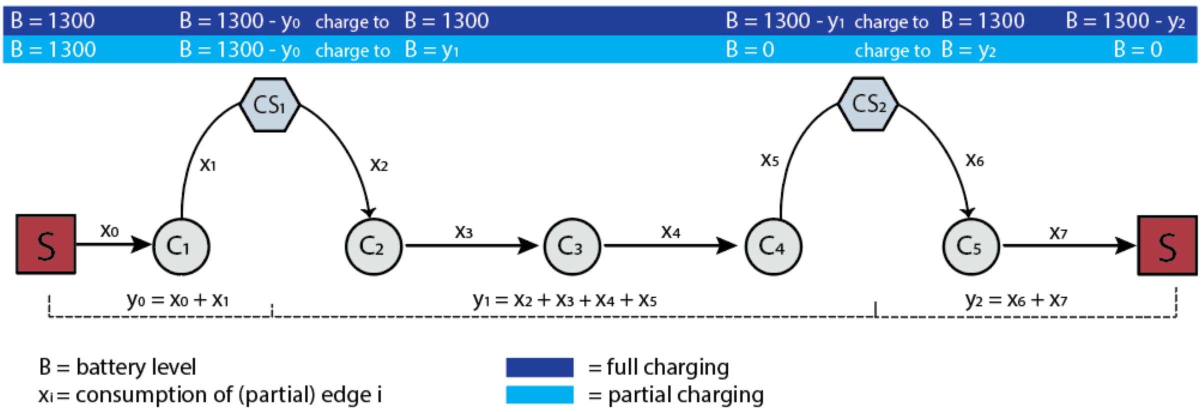


Figure 7: Executing a route when fully or partial charging

However, considering only the cheapest routes of LNS-E2E* and adding corresponding charging time, could prevent choosing the route that has the least total cost. Namely, a tour with more distance cost might have less charging cost and vice versa. Considering partial charging, more CS visits could lead to charging to a lower level at each CS. In this manner, one would avoid the slow charging speed at higher battery levels. Figure 8 illustrates an example of this issue. As a result, selecting the route with the least total cost requires exploring more tours than only the cheapest. But, as many different routes from one satellite to the desired sequence of customers via CSs are possible, determining the cost of every route would take much computational time. Therefore, only routes with a distance and usage cost smaller or equal to uC, where C indicates the total cost of the cheapest tour found by LNS-E2E*, and u refers to a parameter value, are considered to determine the most efficient tour when including charging time. When the total cost of the selection of tours is determined, the one with the least cost is chosen. This process is incorporated when CS visits are inserted in the reconstruction phase of LNS-E2E*, the rest of LNS-E2E* stays the same.

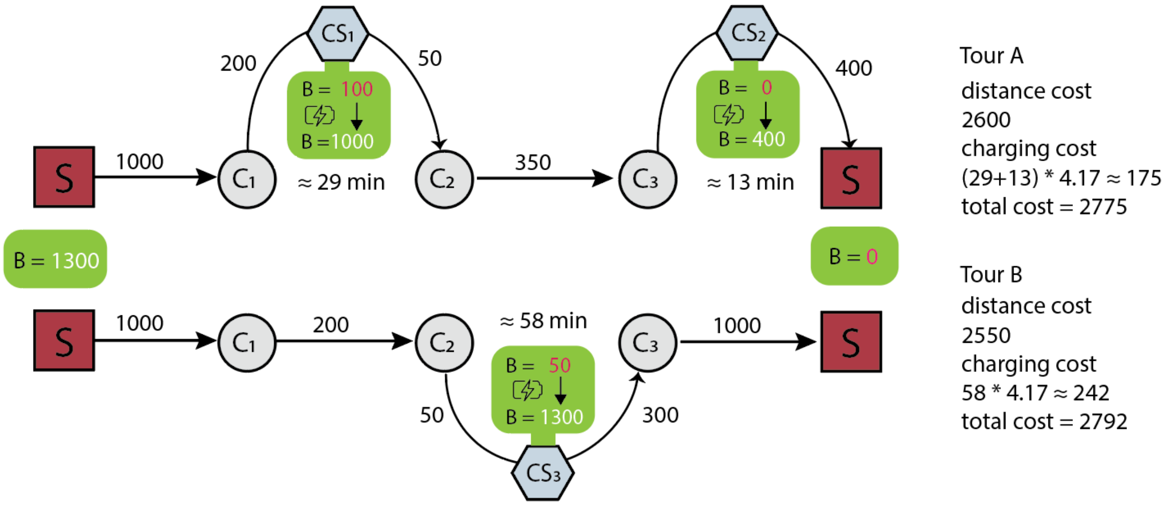


Figure 8: Comparing two tours to select the one with least total cost

6 Computational study

LNS-E2E* was coded in Java Eclipse (JRE 1.7.0-21) and the experiments were executed on an Intel(R) Core(TM) i7-4700MQ CPU, 2.40 Ghz, 8 Gb RAM, running Windows 10. This section presents the computational study of the implemented algorithms. First, additional information of the used instances is given in subsection 6.1. Subsection 6.2 presents the parameter determination and in subsection 6.3 the results of the replication part are shown first, whereafter the results of the extensive part are given.

6.1 Data

The benchmark instances used in this study are a subset of the instance sets used by Breunig et al. (2019) and obtained from their website https://w1.cirrelt.ca/ vidalt/en/VRP-resources.html/. For the extensive part, only the subset of instances from 'Set 8' of Breunig et al. (2019) is used such that the battery capacity of an EV equals L = 1300. The non-linear charging function of the moderate charging type with its characteristics is adopted from Montoya et al. (2017). The average speed of the EV when performing a tour is adopted from Felipe et al. (2014).

6.2 Parameter settings

To create an accurate replication p_1, p_2, p_3, p_4 and i_{max} were adopted from Breunig et al. (2019) while performing LNS-E2E*. For the extensive part parameter u was added and determined through experimentation. Table 1 presents these values.

Table 1: Parameter values used during executing LNS-E2E* for both with and without charging costs

	Parameter	Value
p_1	Related nodes removal (%)	11
p_2	Random routes removal (%)	37
p_3	Open all satellites (%)	12
p_4	Remove single customer routes (%)	18
i_{max}	Number of non-improving iterations before restart	385
u	Selecting routes to integrate charging costs	1.1

6.3 Numerical Results

6.3.1 LNS-E2E*

Seventeen sets of instances are considered to evaluate the performance of LNS-E2E*. Four sets are small sizes, that is with 21 customers, nine sets are medium-sized; with 32, 50, or 75 customers and four sets have a large size with 100 or 200 customers. In all of the instances, the fixed usage cost of the vehicles equals zero and the consumption per distance unit equals 1.0. Table 2 presents the average and best cost for both LNS-E2E and LNS-E2E* for small and medium-size instances, and the gap between a) the two averages, b) the average of LNS-E2E* and the optimal solution and, c) the optimal solution and best result of LNS-E2E*. Cargo capacities Q_1 and Q_2 coincide for each set with the same number of customers. For 21 customers: Q_1 = 15000 and Q_2 = 6000, for 32 customers: Q_1 = 20000 and Q_2 = 8000, for 50 customers Q_1 = 640 and Q_2 = 160 and for 75 customers Q_1 = 560 and Q_2 = 140.

For the large-size instances, only the gap between the averages is presented as no optimal solution was found by Breunig et al. (2019). The characteristics of the small and medium-size instances included in Table 2 are the number of customers n_c , satellites n_s , trucks m^1 , EVs m^1, CSs n_r and battery capacity L. For the large size instances also Q_1 and Q_2 are presented as they differ per set, which is shown in Table 3. The performance of LNS-E2E* is evaluated by comparing the results with the results of LNS-E2E from Breunig et al. (2019). The results were obtained by running the LNS-E2E* algorithm ten times, with for each time running LNS-E2E* the same stopping condition as Breunig et al. (2019). That is, for small and medium-size instances after 150 seconds and, for large size instances after 15 minutes.

$\text{T}_{max} = 150s$	Characteristics						**LNS-E2E**		**LNS-E2E***				
Instance	n_c	n_s	m^1	m^2	n_r	L	Avg	Best	Avg	Best	Avg Gap	Avg Op Gap	Opt Gap
Small size instances													
Set2-n22-k4-s6-17	21	2	3	4	4	470	**5229.0**	**5229**	5524.8	5382	5.64%	5.64%	2.93%
Set2-n22-k4-s8-14	21	2	3	4	4	600	5168.4	**5094**	5166.9	**5094**	-0.03%	1.43%	0.00%
Set3-n22-k4-s13-14	21	2	3	4	4	660	6406.8	**6396**	6572.2	6408	2.58%	2.75%	0.19%
Set3-n22-k4-s13-16	21	2	3	4	4	610	6954.2	**6922**	7130.8	6933	2.54%	3.02%	0.16%
Medium size instances													
Set3-n33-k4-s16-22	32	2	3	4	6	810	7656.2	**7561**	7833.6	7694	2.32%	3.61%	1.76%
Set3-n33-k4-s16-24	32	2	3	4	6	840	7520.0	**7501**	7976.9	7829	6.08%	6.34%	4.37%
Set3-n33-k4-s19-26	32	2	3	4	6	790	7223.2	**7212**	7524.3	7364	4.17%	4.33%	2.11%
Set6a-n51-4	50	4	2	50	5	610	7879.4	**7663**	8931.8	8772	13.36%	16.56%	14.47%
Set6a-n51-5	50	5	2	50	6	490	8386.4	**8288**	8841.4	8721	5.43%	6.68%	5.22%
Set6a-n51-6	50	6	2	50	7	490	7943.8	**7795**	8331.0	8177	4.87%	6.88%	4.90%
Set6a-n76-4	75	4	3	75	7	600	10692.0	**10599**	11055.1	10814	3.40%	4.30%	2.03%
Set6a-n76-5	75	5	3	75	7	500	11242.4	**11178**	11993.3	11751	6.68%	7.29%	5.13%
Set6a-n76-6	75	6	3	75	7	630	10250.0	**10156**	10972.7	10441	7.05%	8.04%	2.81%

Table 2: Numerical results of LNS-E2E and LNS-E2E* for small and medium size instances

For the small-size instances, the results of LNS-E2E* match the results of LNS-E2E in only one of the instance sets: Set2-n22-k4-s8-14, which can be explained partly by the Local Search algorithms used in LNS-E2E for further improvements of the routes. Because LNS-E2E* does not incorporate these algorithms, the tour schedule is not optimized further after each reconstruction phase. In three out of four small-size instances, the best results obtained from LNS-E2E* had an optimality gap of less than 0.2%. The EVs of these three instance sets had all a battery capacity of at least $L = 600$. The instance set that performed the least accurately is characterized with $L = 470$. The lower battery capacity might have been the cause of the performance difference. Taking the average of the gaps between the average results of the small instances and their optimal solution yields an average gap of 3.21%. So, even without the Local Search algorithms, LNS-E2E* can still achieve accurate results for small-size instances.

For the medium-size instances, the accuracy of LNS-E2E* decreases. For the instance sets with 32 customers, LNS-E2E* produces on average results with an optimality gap of 4.76%, which is still quite small. From the instances with 50 customers, the average optimality gap is 10.04%. Especially for Set6a-n51-4, LNS-E2E* did not achieve near-optimal results. The difference between this set from the other sets of 50 customers was a lower number of satellites and CSs along with a higher EV battery capacity. It seems therefore more difficult for LNS-E2E* to find near-optimal solutions when fewer satellites and fewer CSs are included in the E2E-VRP. For the instance sets with 75 customers, the average optimality gap is 6.54%. Remarkable is that, on average, LNS-E2E* achieved better results for the instances with 75 customers than with 50 customers, which is contrary to the expected performance of LNS-E2E*. Namely, one would expect, through the NP-hard character of the problem, that the performance of LNS-E2E* reduces when problem size increases. However, looking at the sets separately and comparing the sets of 50 and 75 customers with the same number of satellites, better performance for a larger size instance only appeared once. And, because the average gap of Set6a-n51-4 is very high compared to the other gaps of medium size instance sets, it could be an exceptional case or might be the result of having a small number of CSs. Thus, it is hard to say whether LNS-E2E* performs better for instance sets with 50 or 75 customers. Appendix A present two examples of a tour schedule for the instance sets 3a-n33-k4-s16-24 and 6a-n76-5.

$T_{max} = 900s$	Characteristics								**LNS-E2E**		**LNS-E2E***		
Instance	n_c	n_s	m^1	m^2	n_r	Q_1	Q_2	L	Avg	Best	Avg	Best	Avg Gap
Large size instances													
Set5-100-5-1	100	5	5	32	10	528	70	460	16224.6	16167	19569.3	19384	20.61%
Set5-100-5-1b	100	5	5	15	10	528	150	490	12070.2	11937	14198.1	13904	17.63%
Set5-200-10-1	200	10	5	62	20	1033	70	280	16354.8	16016	20793.1	20226	27.14%
Set5-200-10-1b	200	10	5	30	20	1033	150	270	12975.4	12771	15989.7	15718	23.23%

Table 3: Numerical results of LNS-E2E and LNS-E2E* for large size instances

For large size instance sets, the accuracy of LNS-E2E* reduces more. The average gap between the results of LNS-E2E and LNS-E2E* is 22.15%, which is much larger than the average gap associated with the medium-sized instance sets. Also here, we observe diminishing accuracy of LNS-E2E* when the problem sizes increase. Several factors could cause the large gaps. As well as for the small and medium-size instances, the absence of the Local Search algorithms might have a big influence on the obtained solutions. Moreover, as time was the stopping condition and Breunig et al. (2019) used a different processor (3.4 GHz Intel i7-3770 CPU), LNS-E2E could have had more iterations than LNS-E2E*, creating a higher chance to obtain better solutions. The different processor has probably a larger impact for the large size instances as the stopping condition was after 15 minutes instead of only 150 seconds and the relative difference in iterations finished by LNS-E2E* or LNS-E2E becomes much bigger for a larger time. The performance of LNS-E2E* seems better when including fewer EVs with higher cargo capacity Q_2.

6.3.2 Full and partial charging

Five sets of medium size instances with 50 customers were used to evaluate the results of full and partial charging. For all the sets in Table 4, the cargo capacity of trucks, Q_1, and of EVs, Q_2, and the battery capacity L were 250, 125, and 1300 respectively. The results were obtained from ten runs for each instance set. Each run was terminated after ten minutes. Table 4 shows the average total cost, cheapest cost found, and the average number of visits to CSs for full and partial charging. The consumption per distance unit equals 1.0 and, the fixed usage cost of the vehicles equals zero for each set. The total cost involves the distance and charging cost. The right-most columns present the relative difference between the average results and the best results that were obtained when partial charging was allowed. In all of the test instances, the relative difference was less than 10%. On average, partial charging reduces the total cost of full charging by 7.3%. In four out of five sets, the number of CS visits increases when partial charging is allowed. However, Set8-05-1300 shows the opposite as the number of CS visits decreases when partial charging is allowed. On average the number of CS visits in a tour schedule for fully charging increases from 8.7 to 9.1 when partial charging is allowed. This increase could indicate that it might be more efficient to charge more frequently to a lower level. Yet, more observations are needed to see if this increase is significant and to formulate a reliable conclusions that rely on statistical tests. Looking at the relative differences between the best results of full and partial charging, partial charging achieves a reducement of 6.1%, on average.

$\mathrm{T}_{max} = 600s$	Characteristics					**Full**	**Charge**		**Partial**	**Charge**			
Instance	n_c	n_s	m^1	m^2	n_r	Avg	Best	Avg CS	Avg	Best	Avg CS	Avg dif	Dif best
Set8-01-1300	50	4	6	10	20	18132.0	16745	9.9	16606.2	16169	11.8	-9.2%	-3.9%
Set8-02-1300	50	4	6	10	20	18088.5	17442	8.0	17137.9	16104	9.3	-5.5%	-8.3%
Set8-03-1300	50	4	6	10	20	16954.6	15711	8.4	15623.8	15041	8.8	-8.5%	-4.5%
Set8-04-1300	50	4	6	10	20	19218.0	18605	7.1	17993.8	17429	7.6	-6.8%	-6.7%
Set8-05-1300	50	4	6	10	20	16815.6	16382	10.0	15789.0	15310	7.9	-6.5%	-7.0%

Table 4: Total cost for full and partial charging

7 Conclusions and future research

In this paper, the effect of integrating a non-linear charging function and allowing for partial charging in the Electric Two-Echelon Vehicle Routing Problem was investigated. This was achieved by replicating the proposed metaheuristic LNS-E2E from Breunig et al. (2019), which resulted in LNS-E2E*, a modified version of LNS-E2E. Because LNS-E2E* is less advanced than LNS-E2E, LNS-E2E* achieved less accurate results than were found by Breunig et al. (2019). Still, LNS-E2E* achieved good results for small-size instances, but the accuracy of the heuristic diminished when the problem size of the instances increased.

After the replication, a method was proposed to modify LNS-E2E* such that charging costs are taken into account with the determination of efficient routes. When partial charging was allowed, an average reducement of 7.3% of the total cost was found. The average number of CS visits increased for partial charging, from 8.7 CS visits in a tour schedule to 9.1 CS visits. This could indicate that planning CS visits more frequently could result in less total cost to avoid the slow charging rate at a high battery level. Still, because this finding does not rely on statistical tests, more observations are needed to determine if this conjecture is true.

Thus, to formulate an answer to the research question, from the obtained results, we observe that incorporating partial charging and a non-linear charging function provide a lower total cost compared to fully charging. Also, a slight increase in CS visits was found when partial charging was allowed, which can be explained by the decreasing charging speed. Hence, it could be beneficial for companies to consider partial recharging and incorporate this in the planning as the total cost will be less than for full charging.

The opportunities for future research are multiple. This study was the first to incorporate charging time and a non-linear charging function for the determination of efficient routes. Still, many different aspects of the E2E-VRP could be investigated further. First of all, this study considered only one charging type and, it would be interesting to examine the effect of having multiple charging types at each CS. Secondly, this study assumes deterministic consumption for each edge, which makes predetermining the recharge amount straightforward. However, in the real world, consumption is not deterministic as vehicles could, for example, get stuck in traffic. Therefore, incorporating stochasticity and uncertainty would make the problem more realistic. A third opportunity is to integrate waiting for time and availability of CSs as companies do not always have a private network of CSs and use shared CSs. Furthermore, it would be interesting to extend the proposed method for integrate charging costs. The method in this study only considered the consumption of distinct segments of a route. But, it could be more efficient to consider the whole route and allocate charging amounts to CS visits, such that they could differ from the amount allocated in this study.

8 Discussion

With the evaluation of the results and the establishment of the conclusions, it is important to be aware of the limitations of this research. First of all, the results of LNS-E2E* were obtained from only ten runs per instance set. More runs would create a more justified conclusion that would rely on statistically verified arguments. Now, conclusions only rely on visually obtained differences, which is not reliable. This argument also applies to the results of the extensive part. Due to the limited time of this research, too little observations were attained and, conclusions based on conjectures were formulated. So, it would be convenient to create more runs and examine corresponding results. Furthermore, the results of LNS-E2E* were obtained with a different processor that was slower than the processor used by Breunig et al. (2019). So comparing the results of LNS-E2E* and LNS-E2E, when time is used as a stopping condition, is not reliable. Additionally, the proposed method to integrate charging costs is not optimal. So, when this method would be more advanced, the difference between full and partial charging could have been larger and different results could be achieved. Also for the extensive part, the stopping condition could have been insufficient. Because the algorithm for the extensive part needs much more computational time than the LNS-E2E* without charging costs, 10 minutes might be too short to achieve near optimal solutions and the obtained results could have been different when the stopping condition would be for example 20 minutes. Due to the limited time of this research, the stopping condition was determined without comparing different stopping criteria.

References

Breunig, U., Baldacci, R., Hartl, R. & Vidal, T. (2019), 'The electric two-echelon vehicle routing problem', *Computers Operations Research* **103**, 198–210.
URL: *https://www.sciencedirect.com/science/article/pii/S0305054818302909*

Dantzig, G. B. & Ramser, J. H. (1959), 'The truck dispatching problem', *Management Science* **6**(1), 80–91.
URL: *http://www.jstor.org/stable/2627477*

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', **71**, 111–128.

Gonzalez-feliu, J., Perboli, G., Tadei, R. & Vigo, D. (2008), 'The two-echelon capacitated vehicle routing problem To cite this version : The Two-Echelon Capacitated Vehicle Routing Problem', *<Halshs-00879447>* .

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 and Operations Research* **39**(12), 3215–3228.

Kancharla, S. R. & Ramadurai, G. (2020), 'Electric vehicle routing problem with non-linear charging and load-dependent discharging', *Expert Systems with Applications* **160**, 113714.
URL: *https://doi.org/10.1016/j.eswa.2020.113714*

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

Lin, J., Zhou, W. & Wolfson, O. (2016), 'Electric Vehicle Routing Problem', *Transportation Research Procedia* **12**(June 2015), 508–521.

Miller-hooks, E. & Erdogan, S. (2012), 'A Green Vehicle Routing Problem', **48**(2012), 100–114.

Montoya, A., Guéret, C., Mendoza, E. & Villegas, G. J. (2017), 'The electric vehicle routing problem with nonlinear charging function', *Transportation Research Part B: Methodological* **103**, 87–110.

Sweda, T. M., Dolinskaya, I. S. & Klabjan, D. (2017), 'Adaptive routing and recharging policies for electric vehicles', *Transportation Science* **51**(4), 1326–1348.

Appendix A

Example of a tour from Set3a-n33-k4-s16-24

total cost is 7829

total tours first level is 2

Best First Level Tour Schedule with cost 1070

FLT 1 with commodity 15370 and cost 890

depot: (2600, 3750) –>sat: (3040, 3820) –>depot: (2600, 3750)

FLT 2 with commodity 14000 and cost 180

depot: (2600, 3750) –>sat: (2610, 3840) –>depot: (2600, 3750)

Best SecondLevelTour Schedule with cost 6759 and 4 tours:

Tour 1

total commodity is 7920, total cost of the tour is 2302

sequence of customers

Sat (3040,3820) D=15090 -> Cus (3140,4350) D=1500 -> Cus (3110,4420) D=150 -> Cus (3090,4450) D=400 -> Cus (3070,4640) D=400 -> Cus (3360,4750) D=1200 -> Cus (3200,4390) D=40 -> Cus (3210,4370) D=80 -> Cus (3220,4370) D=2000 -> Cus (3150,4070) D=650 -> Cus (3140,4060) D=200 -> Cus (3140,3940) D=1300 -> Sat (3040,3820) D=15090

tour CS

satellite (3040, 3820) –539–>customer (3140, 4350) –76–>customer (3110, 4420) –36–>customer (3090, 4450) – CS: (3120,4580) with cost: 211 and cons: 78.0 –>customer (3070, 4640) –310–>customer (3360, 4750) – CS: (3120,4580) with cost: 500 and cons: 206.0 –>customer (3200, 4390) –22–>customer (3210, 4370) –10–>customer (3220, 4370) –308–>customer (3150, 4070) –14–>customer (3140, 4060) –120–>customer (3140, 3940) –156–>satellite (3040, 3820)

Tour 2

total commodity is 6400, total cost of the tour is 837

sequence of customers

Sat (2610,3840) D=14280 -> Cus (2950,4020) D=1400 -> Cus (2830,4060) D=4000 -> Cus (2710,4010) D=1000 -> Sat (2610,3840) D=14280

tour CS

satellite (2610, 3840) –384–>customer (2950, 4020) –126–>customer (2830, 4060) –130–>customer

(2710, 4010) –197–>satellite (2610, 3840)

Tour 3

total commodity is 7450, total cost of the tour is 1497

sequence of customers

Sat (3040,3820) D=15090 -> Cus (3040,3820) D=750 -> Cus (3130,3780) D=700 -> Cus (3210,3910) D=400 -> Cus (3210,3980) D=300 -> Cus (3230,4290) D=750 -> Cus (3240,4330) D=600 -> Cus (3230,4330) D=900 -> Cus (3190,4330) D=1100 -> Cus (3040,4270) D=250 -> Cus (2980,4270) D=700 -> Cus (2960,4180) D=450 -> Cus (2970,4100) D=550 -> Sat (3040,3820) D=15090

tour CS

satellite (3040, 3820) –0–>customer (3040, 3820) –98–>customer (3130, 3780) –152–>customer (3210, 3910) –70–>customer (3210, 3980) –310–>customer (3230, 4290) – CS: (3300,4320) with cost: 136 and cons: 60.0 –>customer (3240, 4330) –10–>customer (3230, 4330) –40–>customer (3190, 4330) –161–>customer (3040, 4270) –60–>customer (2980, 4270) –92–>customer (2960, 4180) –80–>customer (2970, 4100) –288–>satellite (3040, 3820)

Tour 4

total commodity is 7600, total cost of the tour is 2123

sequence of customers

Sat (2610,3840) D=14280 -> Cus (2610,3840) D=700 -> Cus (2790,3990) D=600 -> Cus (2930,4210) D=1600 -> Cus (2900,4340) D=1700 -> Cus (2770,4390) D=2500 -> Cus (2640,4140) D=500 -> Sat (2610,3840) D=14280

tour CS

satellite (2610, 3840) –0–>customer (2610, 3840) –234–>customer (2790, 3990) – CS: (3040,3820) with cost: 707 and cons: 405.0 –>customer (2930, 4210) –133–>customer (2900, 4340) – CS: (2760,4580) with cost: 467 and cons: 190.0 –>customer (2770, 4390) –281–>customer (2640, 4140) –301–>satellite (2610, 3840)

Example of a tour for Set6a-n76-5

total cost is 11909

total tours first level is 3

Best First Level Tour Schedule with cost 3442

FLT 1 with commodity 349 and cost 1020

depot: (10, 10) –>sat: (480, 210) –>depot: (10, 10)

FLT 2 with commodity 487 and cost 1728

depot: (10, 10) –>sat: (220, 530) –>sat: (550, 570) –>sat: (540, 380) –>depot: (10, 10)

FLT 3 with commodity 528 and cost 694

depot: (10, 10) –>sat: (270, 240) –>depot: (10, 10)

Best SecondLevelTour Schedule with cost 8467 and 11 tours:

Tour 1

total commodity is 105, total cost of the tour is 363

sequence of customers

Sat (550,570) D=0 -> Cus (550,570) D=22 -> Cus (500,700) D=9 -> Cus (570,720) D=37 -> Cus (550,650) D=37 -> Sat (550,570) D=0

tour CS

satellite (550, 570) –0–>customer (550, 570) –139–>customer (500, 700) –72–>customer (570, 720) –72–>customer (550, 650) – CS: (550,570) with cost: 80and cons: 0.0 –>satellite (550, 570)

Tour 2

total commodity is 123, total cost of the tour is 1329

sequence of customers

Sat (480,210) D=273 -> Cus (550,200) D=21 -> Cus (600,150) D=14 -> Cus (660,140) D=22 -> Cus (660,80) D=11 -> Cus (640,40) D=13 -> Cus (590,50) D=3 -> Cus (500,40) D=8 -> Cus (540,100) D=12 -> Cus (500,150) D=19 -> Sat (480,210) D=273 *tour CS*

satellite (480, 210) –70–>customer (550, 200) –70–>customer (600, 150) –60–>customer (660, 140) –60–>customer (660, 80) – CS: (480,210) with cost: 455 and cons: 233.0 –>customer (640, 40) –50–>customer (590, 50) – CS: (480,210) with cost: 365 and cons: 171.0 –>customer (500, 40) –72–>customer (540, 100) –64–>customer (500, 150) –63–>satellite (480, 210)

Tour 3

total commodity is 134, total cost of the tour is 781

sequence of customers

Sat (270,240) D=274 -> Cus (260,130) D=12 -> Cus (150,50) D=28 -> Cus (150,140) D=11 -> Cus (120,170) D=15 -> Cus (60,250) D=26 -> Cus (110,280) D=6 -> Cus (160,190) D=18 -> Cus (220,220) D=18 -> Sat (270,240) D=274

tour CS

satellite (270, 240) –110–>customer (260, 130) –136–>customer (150, 50) –90–>customer (150, 140) –42–>customer (120, 170) – CS: (60,190) with cost: 123 and cons: 60.0 –>customer (60, 250) –58–>customer (110, 280) –102–>customer (160, 190) –67–>customer (220, 220) –53–>satellite (270, 240)

Tour 4

total commodity is 118, total cost of the tour is 597

sequence of customers

Sat (540,380) D=419 -> Cus (500,400) D=19 -> Cus (510,420) D=27 -> Cus (550,450) D=16 -> Cus (500,500) D=15 -> Cus (550,500) D=10 -> Cus (620,480) D=15 -> Cus (670,410) D=16 -> Sat (540,380) D=419

tour CS

satellite (540, 380) –44–>customer (500, 400) –22–>customer (510, 420) – CS: (540,380) with cost: 120 and cons: 70.0 –>customer (550, 450) –70–>customer (500, 500) –50–>customer (550, 500) –72–>customer (620, 480) –86–>customer (670, 410) –133–>satellite (540, 380)

Tour 5

total commodity is 126, total cost of the tour is 743

sequence of customers

Sat (550,570) D=0 -> Cus (700,640) D=24 -> Cus (620,570) D=31 -> Cus (400,600) D=21 -> Cus (400,660) D=26 -> Cus (470,660) D=24 -> Sat (550,570) D=0

tour CS

satellite (550, 570) –165–>customer (700, 640) –106–>customer (620, 570) – CS: (550,570) with cost: 222 and cons: 152.0 –>customer (400, 600) –60–>customer (400, 660) –70–>customer (470, 660) –120–>satellite (550, 570)

Tour 6

total commodity is 126, total cost of the tour is 653

sequence of customers

Sat (270,240) D=274 -> Cus (300,200) D=18 -> Cus (350,160) D=29 -> Cus (360,60) D=15 -> Cus (440,130) D=28 -> Cus (400,200) D=10 -> Cus (360,260) D=26 -> Sat (270,240) D=274

tour CS

satellite (270, 240) –50–>customer (300, 200) –64–>customer (350, 160) –100–>customer (360, 60) –106–>customer (440, 130) – CS: (480,210) with cost: 169 and cons: 80.0 –>customer (400, 200) –72–>customer (360, 260) –92–>satellite (270, 240)

Tour 7

total commodity is 124, total cost of the tour is 568

sequence of customers

Sat (480,210) D=273 -> Cus (480,210) D=20 -> Cus (620,240) D=8 -> Cus (650,270) D=14 -> Cus (620,350) D=12 -> Cus (540,380) D=19 -> Cus (550,340) D=17 -> Cus (500,300) D=21 -> Cus (520,260) D=13 -> Sat (480,210) D=273

tour CS

satellite (480, 210) –0–>customer (480, 210) –143–>customer (620, 240) –42–>customer (650, 270) –85–>customer (620, 350) –85–>customer (540, 380) – CS: (540,380) with cost: 41 and cons: 41.0 –>customer (550, 340) –64–>customer (500, 300) –44–>customer (520, 260) – CS: (480,210) with cost: 64and cons: 0.0 –>satellite (480, 210)

Tour 8

total commodity is 138, total cost of the tour is 1281

sequence of customers

Sat (270,240) D=274 -> Cus (270,240) D=6 -> Cus (200,300) D=11 -> Cus (120,380) D=5 -> Cus (70,430) D=27 -> Cus (210,450) D=11 -> Cus (210,480) D=17 -> Cus (300,600) D=16 -> Cus (350,600) D=1 -> Cus (350,510) D=16 -> Cus (410,460) D=18 -> Cus (380,330) D=10 -> Sat (270,240) D=274

tour CS

satellite (270, 240) –0–>customer (270, 240) –92–>customer (200, 300) –113–>customer (120, 380) –70–>customer (70, 430) –141–>customer (210, 450) –30–>customer (210, 480) – CS: (220,530) with cost: 156 and cons: 106.0 –>customer (300, 600) –50–>customer (350, 600) –90–>customer (350, 510) –78–>customer (410, 460) – CS: (540,380) with cost: 319 and cons: 167.0 –>customer (380, 330) – CS: (270,240) with cost: 142and cons: 0.0 –>satellite (270, 240)

Tour 9

total commodity is 102, total cost of the tour is 467

sequence of customers

Sat (480,210) D=273 -> Cus (430,260) D=22 -> Cus (400,370) D=20 -> Cus (450,420) D=30 -> Cus (450,350) D=30 -> Sat (480,210) D=273 *tour CS*

satellite (480, 210) –70–>customer (430, 260) –114–>customer (400, 370) –70–>customer (450, 420) –70–>customer (450, 350) –143–>satellite (480, 210)

Tour 10

total commodity is 130, total cost of the tour is 702

sequence of customers

Sat (270,240) D=274 -> Cus (330,340) D=19 -> Cus (290,390) D=12 -> Cus (330,440) D=20 -> Cus (300,500) D=33 -> Cus (210,360) D=19 -> Cus (260,290) D=27 -> Sat (270,240) D=274

tour CS

satellite (270, 240) –116–>customer (330, 340) –64–>customer (290, 390) –64–>customer (330, 440) –67–>customer (300, 500) – CS: (220,530) with cost: 255 and cons: 170.0 –>customer (210, 360) –86–>customer (260, 290) –50–>satellite (270, 240)

Tour 11

total commodity is 138, total cost of the tour is 983

sequence of customers

Sat (220,530) D=398 -> Cus (220,530) D=28 -> Cus (260,590) D=29 -> Cus (310,760) D=25 -> Cus (170,640) D=14 -> Cus (100,700) D=7 -> Cus (90,560) D=13 -> Cus (150,560) D=22 -> Sat (220,530) D=398

tour CS

satellite (220, 530) –0–>customer (220, 530) –72–>customer (260, 590) –177–>customer (310, 760) – CS: (220,530) with cost: 366 and cons: 120.0 –>customer (170, 640) –92–>customer (100, 700) –140–>customer (90, 560) –60–>customer (150, 560) – CS: (220,530) with cost: 76and cons: 0.0 –>satellite (220, 530)

Appendix B

The written code is submitted via SIN-online in a ZIP file. A short description of the code is given below: The 'main' reads the instance files and calls the LNS-E2E* method: applyLNSE2E. The 'applyLNSE2E' method performs all the calls needed to execute the LNS-E2E* algorithm. Several classes are initialized for different sets with their corresponding characters, such as trucks, EVs, customers, charging stations, satellites, etc. Within the applyLNSE2E-method, one can distinguish the LNS-E2E* for the replicative and extensive part. For the replication, the AddCS method is called to apply the DP algorithm and find feasible solutions for the battery capacity. For the extensive part, the 'AddCSAndCTFull'and 'AddCSAndCTPartial' methods solve the E2E-VRP with the charging costs for full and partial recharging respectively.