Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operational Research

The Green Mixed Fleet Vehicle Routing Problem with Steep Routes, Partial Battery Recharging and Time Windows

Nienke Kempes (537197)

Supervisor:	Y.N. Hoogendoorn
Date final version:	2nd July 2023

Abstract

This thesis includes the effects of road gradient on CO_2 emission and energy consumption in the proposed Green-Vehicle Routing Problem with Steep Routes, time windows, partial battery recharge and a mixed fleet of conventional and electrical vehicles. We investigate at which levels of road gradients we should include road gradients the model. We use test instances that consist of a large number of customers. Furthermore, we use an iterated local search metaheuristic to solve the problem, where we make use of inter local search techniques. The results show significant cost reductions, especially when the height differences between customers are greater than 100 meters.

1 Introduction

CO_2 emissions are a significant problem in the transport industry. According to the International Energy Agency, global transport still accounts for 24% of total CO_2 emissions and 29.4% of the global transport emissions are caused by trucks (Agency, 2018). It is therefore important to reduce these CO_2 emissions in the transport industry. The amount of CO_2 that a vehicle emits per kilometer mainly depends on the speed of the vehicle, the total weight of the vehicle and the differences in altitude that the vehicle has to climb or descend. So, it is important to take these factors into account when designing efficient transport and distribution systems. The aim of this thesis is to present the Vehicle Routing Problem (VRP), this is an optimization problem that finds the optimal set of routes for a fleet of vehicles, with a green perspective with a mixed fleet of vehicles, conventional vehicles and electrical vehicles, where we include partial battery recharging for electrical vehicles and time windows. The VRP with time windows is a frequently faced problem by several transport companies where customers must be served within a given time interval. Transport companies aim to reduce the costs and on the one hand, electric vehicles do not emit CO_2 and conventional vehicles emit CO_2, but on the other hand electrical vehicles are more expensive than conventional vehicles. So, we will include a mixed fleet to make an optimal balanced decision. Furthermore, because the battery capacities of electrical vehicles are very low and distances can be very large, we allow partial battery recharging to travel routes with large distances. However, recent papers on these topics omit the altitude differences between customers when calculating CO_2 emissions and energy consumption. However, this can be important for areas with high altitude differences as the CO_2 emission per km kan vary. Energy consumption per kilometer can vary between close to 0 kWh per kilometer travelled and 1.5 kWh per kilometer in areas with high road gradients (Liu et al., 2017). Furthermore, for fuel consumption, in areas with very steep routes, fuel consumption can very between 0.1 liter fuel used per kilometer and 1 liter fuel used per kilometer (Zhang et al., 2015). So with large distances between customers in areas with very steep routes, it is important to incorporate these factors. We will investigate from what level of altitude difference between all customers it is significant to include altitude differences.

We call this problem the G-VRP with Steep Routes (G-VRPSR), we will use the test instances from Macrina et al. (2019), which we will adapt by associating uniform randomly generated elevation information on the nodes.

The remainder of this thesis is structured as follows. In Section 2, we will give a short review of the related scientific literature. Section 3 provides a detailed description of the G-VRPSR. Solution algorithms for the G-VRP and G-VRPSR are described in Section 4. The computational experiments are reported Section 5. Finally, Section 6 discusses the outcomes and potential future research.

2 Literature review

The Vehicle Routing Problem (VRP) is a well-known optimization problem that optimizes the costs for the routes for multiple vehicles in order to deliver a given set of customers. Dantzig and Ramser (1959) were the first to introduce this problem are made. Nowadays, many extensions for this problem. For example, Solomon (1987) introduced the Vehicle Routing Problem with Time Windows (VRP-TW) and Min (1989) introduced the Vehicle Routing Problem with simultaneous delivery and pick-up points. However, in recent years, there has been an increased interest in the pollution and sustainability aspects of the VRP, and thus the Green VRPs (G-VRPs) were introduced. G-VRPs are special in the fact that they include limiting CO_2 emission in minimizing the route costs by minimizing including CO_2 emission in the objective or by setting an upper bound on CO_2 emission emitted. The first that studied the G-VRP were Bektaş and Laporte (2011), who modelled the energy consumption of conventional vehicles and their polluting impact in the Pollution-Routing Problem (PRP), where different parameters for load capacity and vehicle speed where taken into account for computing emissions. As minimizing carbon emissions increases driving time, Demir et al. (2014) introduced a bi-objective PRP for minimizing costs and minimizing carbon emissions. Jabali et al. (2012) solved a Time-Dependent VRP (T-DVRP) by tabu search considering the maximum achievable speed as part of the optimization and showed that reducing CO_2 emissions also leads to reducing operating costs. Tajik et al. (2014) solved the Time Dependent PRP (TDPRP) with uncertain data and with simultaneous delivery and pick-up points. The Electric-Vehicle Routing Problem (E-VRP) was introduced by Lin et al. (2016), this VRP considered the vehicle load effect on battery consumption and included recharge station visits for charging the battery of vehicles. Following this, Montoya et al. (2016) modelled the charging time as an exponential function instead of a linear function. Sassi et al. (2014) introduced a Heterogenous Electric Vehicle Routing Problem with Time Dependent Charging Costs and a Mixed Fleet (HEVRP-TDMF), where customers could be served by either an electrical vehicle (EV), having different battery capacities and operating costs, or a conventional vehicle (CV). Furthermore, it included recharging with time dependent costs. Macrina et al. (2019) extended the G-VRP with a mixed fleet of vehicles, CVs and EVs, with partial recharge stations for EVs. In addition, they incorporated time windows and considered a limit on polluting emissions. They solved the problem by an iterative local search metaheuristic. Yu et al. (2021) developed an adaptive neighborhood search for the green mixed fleet vehicle routing problem of Macrina et al. (2019) with realistic energy consumption and partial recharges.

However, these papers assumed that CO_2 emissions only depended on the distance travelled or vehicle speed, whereas in reality CO_2 emission depend on several more factors. The grade of the road has been included in the calculation of the CO_2 emission (Suzuki, 2011), the vehicle speed (Demir et al., 2012), traffic congestion (Franceschetti et al., 2013) and the driver's driving habit (Bandeira et al., 2013). Brunner et al. (2021) applied the VRP in urban areas with significant altitude differences in a VRP with Steep Routes (VRP-SR). They modelled routing decisions including the impact of road gradients in a fuel consumption cost model. Palmer (2007) presented an integrated routing and CO_2 emission model for freight vehicles and highlighted the role of speed in reducing CO_2 emissions. L. Liu and Lai (2021) studied the Low Carbon Routing Problem (L-CRP) and proposed a Multi Depot VRP (MDVRP) considering fuel consumption optimization under the condition of the latest receiving time of consumers and developed a multi-population fruit fly algorithm to solve the problem. Zhang et al. (2015) incorporated fuel cost, CO_2 emission cost, and vehicle usage cost into the traditional VRP problem and established a L-CRP and developed a tabu search algorithm. Lai et al. (2021) considers a joint pollution-routing and speed optimization problem (PRP-SO) where fuel costs and CO_2 emissions depend on the vehicle speed, arc payloads, and road grades. They pre-calculated the CO_2 emission factors for every arc such that the total CO_2 emission of an arc is only dependent on the vehicle load and vehicle speed at the moment of using that particular arc. Table 1 gives a summary of the main papers that contribute to the G-VRP with steep routes, where * indicates whether a paper includes the specified parameter.

Table 1: Summary of the literature on the G-VRP and its variants

Reference	**Time windows**	**Time dependency**	**Mixed fleet**	**Partial battery recharge**	**Steep routes**
(Bektaş & Laporte, 2011)	*				
(Demir et al., 2012)	*	*			
(Suzuki, 2011)	*				*
(Franceschetti et al., 2013)	*				
(Jabali et al., 2012)		*			
(Demir et al., 2014)	*				
(Macrina et al., 2019)	*		*	*	
(Lai et al., 2021)	*		*		*
(Brunner et al., 2021)					*
(Sassi et al., 2014)		*	*	*	
(Yu et al., 2021)			*	*	
This thesis	*		*	*	*

3 Problem description

In this section, we will provide a formal description of the problem. We introduce the mixed-integer linear programming formulations in Section 3.1. We formulate the emission of CO_2 using a the comprehensive emission model (CMEM) in Section 3.2 and we will formulate the energy consumption model in Section 3.3.

3.1 Mathematical model

We will describe two different mixed-integer linear programming formulations in this section. One that is the same as in Macrina et al. (2019) and that omits the road gradient of routes between customers, which we describe in Section 3.1.1. The modifications on Macrina et al. (2019), for taking road grade in routes into account are described in Section 3.1.2. The main difference between these two models is in defining the fuel consumption for conventional vehicles and energy consumption for electrical vehicles.

3.1.1 The Green Mixed Vehicle Routing Problem With Partial Battery Recharging and Time Windows

The problem is defined based on on the directed completed graph $\mathcal{G}(\mathcal{V}', \mathcal{A})$, where $\mathcal{V}'$ is the set of customers and recharge stations. Let $\mathcal{N}$ be the set of locations and $\mathcal{R}$ the set of recharge stations and $\mathcal{V} = \mathcal{N} \cup \mathcal{R}$. Furthermore, the depot is also in the set of recharge stations and is denoted by s for the start depot and t for the end depot. For allowing multiple visits for each recharge station, we introduce σ copies of the recharge stations so that a recharge station can be visited $(1+\sigma)$ times. Here, $|\mathcal{R}'| = (1+\sigma)|\mathcal{R}|$, and so $\mathcal{V}' = \mathcal{R}' \cup \mathcal{N}$. For every arc $(i,j) \in \mathcal{A}, i \neq j$, d_{ij} is the distance between the customers and t_{ij} is the travel time between the nodes. Each customer $i \in \mathcal{N}$ has a opening time e_i and a closing time l_i, where the vehicle needs to have arrived between these two times and each customer $i \in \mathcal{N}$ has a service time s_i. Each recharge station $i \in \mathcal{R}'$ has a recharging time ρ that is linear to the energy charged at the station and assumed the same for all recharge stations. The mixed fleet consist of conventional vehicles denoted by C and electrical vehicles denoted by E. We assume that there are infinite number of vehicles for both types. Each type of vehicle has a max load capacity, denoted by Q^C and Q^E. The costs for traveling at every arc $(i,j) \in \mathcal{A}, i \neq j$ are c_{ij}^C and c_{ij}^E for conventional vehicles and electrical vehicles respectively. The costs per kWh recharged at a recharge station is denoted by w^r and is assumed to be the same for every recharge station and w^a is denoted as the costs of a full battery that is charged when a electrical vehicle leaves the depot, so $w^a = B^E w^r$, where B^E is the max battery capacity. The coefficient of energy consumption (in kWh/km) is denoted by π and assumed equal for each arc $(i,j) \in \mathcal{A}, i \neq j$. The modelling of the fuel consumption per kilometer and thereby the CO_2 emission $\epsilon(u_i^C)$ per kilometer for each load u_i^C, that depends on the road gradient is explained in Section 3.2.

In order to model the G-VRP and G-VRPSR we define the following decision variables:

- $x_{ij}^C = \begin{cases} 1 & \text{if the CV travels from } i \text{ to } j,\ (i,j) \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$

- $x_{ij}^E = \begin{cases} 1 & \text{if the EV travels from } i \text{ to } j,\ (i,j) \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$

- z_{ij}, the amount of energy available when arriving at node j from the node i (kWh), $(i,j) \in \mathcal{A}$

- g_{ij}, the amount of energy recharged by the EV at node i from traveling to node j (kWh), $i \in \mathcal{R}'$, $j \in \mathcal{V}'$

- τ_j, the arrival time of the vehicle to the node j (h), $j \in \mathcal{V}'$
- u_i^C, the amount of load left in the vehicle after visiting node i (kg), $i \in \mathcal{V}'$
- u_i^E, the amount of load left in the vehicle after visiting node i (kg), $i \in \mathcal{V}'$

A formulation of the G-VRP is given in (1) - (24):

$$\text{Minimize } w^r \sum_{i \in R'} \sum_{j \in V'} g_{ij} + w^a \sum_{j \in V'} x_{sj}^E + \sum_{(i,j) \in A} c_{ij}^E d_{ij} x_{ij}^E + \sum_{(i,j) \in A} c_{ij}^C d_{ij} x_{ij}^C \tag{1}$$

$$\text{subject to } \sum_{j \in V'} (x_{ij}^E + x_{ij}^C) = 1 \qquad i \in \mathcal{N} \tag{2}$$

$$\sum_{j \in V'} x_{ij}^E \leq 1 \qquad i \in \mathcal{R}' \tag{3}$$

$$\sum_{j \in V' \setminus s} x_{ij}^E - \sum_{j \in V' \setminus t} x_{ji}^E = 0 \qquad i \in \mathcal{V}' \tag{4}$$

$$\sum_{j \in V' \setminus s} x_{ij}^C - \sum_{j \in V' \setminus t} x_{ji}^C = 0 \qquad i \in \mathcal{V}' \tag{5}$$

$$\sum_{i \in V', i \neq s \setminus s} x_{si}^E - \sum_{j \in V', j \neq t} x_{jt}^E = 0 \tag{6}$$

$$\sum_{i \in V', i \neq s \setminus s} x_{si}^C - \sum_{j \in V', j \neq t} x_{jt}^C = 0 \tag{7}$$

$$u_j^E \geq u_i^E + q_j x_{ij}^E - Q^E(1 - x_{ij}^E) \qquad i \in \mathcal{V}' \setminus \{s,t\}, j \in \mathcal{V}' \setminus \{s\} \tag{8}$$

$$u_j^C \geq u_i^C + q_j x_{ij}^C - Q^C(1 - x_{ij}^C) \qquad i \in \mathcal{V}' \setminus \{s,t\}, j \in \mathcal{V}' \setminus \{s\} \tag{9}$$

$$u_j^E \leq Q^E \qquad j \in \mathcal{V}' \tag{10}$$

$$u_j^C \leq Q^C \qquad j \in \mathcal{V}' \tag{11}$$

$$u_s^E = 0 \tag{12}$$

$$u_s^C = 0 \tag{13}$$

$$\tau_j \geq \tau_i + (t_{ij} + s_i) x_{ij}^E - M(1 - x_{ij}^E) \qquad i \in \mathcal{N}, j \in \mathcal{V}' \tag{14}$$

$$\tau_j \geq \tau_i + (t_{ij} + s_i) x_{ij}^C - M(1 - x_{ij}^C) \qquad i \in \mathcal{N}, j \in \mathcal{V}' \tag{15}$$

$$\tau_j \geq \tau_i + t_{ij} x_{ij}^E + \frac{1}{\rho_i} g_{ij} - M(1 - x_{ij}^E) \qquad i \in \mathcal{R}', j \in \mathcal{V}' \tag{16}$$

$$e_j \leq \tau_j \leq l_j \qquad j \in \mathcal{V}' \tag{17}$$

$$z_{ij} \leq (z_{hi} + g_{ij}) - \pi d_{ij} x_{ij}^E + M(1 - x_{ij}^E) + M(1 - x_{hi}^E) \quad h \in \mathcal{V}', i \in \mathcal{V}' \setminus s, j \in \mathcal{V}', i \neq j, i \neq h, j \neq h \tag{18}$$

$$z_{sj} \leq B^E - \pi d_{sj} x_{sj}^E + M(1 - x_{sj}^E) \quad j \in \mathcal{V}' \tag{19}$$

$$g_{ij} \leq B^E - z_{hi} + M(1 - x_{ij}^E) + M(1 - x_{hi}^E) \quad i \in \mathcal{R}' \setminus s, h \in \mathcal{V}', j \in \mathcal{V}' \tag{20}$$

$$z_{ij} \geq 0.1 B^E \quad i \in R', j \in \mathcal{V}' \tag{21}$$

$$g_{ij} \leq 0.9 B^E \quad i \in R', j \in \mathcal{V}' \tag{22}$$

$$\sum_{(i,j) \in A} \epsilon(u_i^C) d_{ij} x_{ij}^C \leq UB \tag{23}$$

$$x_{ij}^E, x_{ij}^C \in \{0,1\}, i \in \mathcal{V}', j \in \mathcal{V}'; u_i^E, u_i^C, \tau_i \geq 0, i \in \mathcal{V}'; g_{ij}, z_{ij} \geq 0, i \in \mathcal{R}', j \in \mathcal{V}'. \tag{24}$$

In this formulation, the objective is to minimize travel costs and recharging costs. Constraint (2) ensures that all customers are visited once by a vehicle. Constraint (3) means that every recharge station can be visited at most once. Furthermore, Constraint (4) and (5) ensure that the inflow of vehicles for every customer and recharge station is the same as the outflow of vehicles and Constraint (6) and (7) ensure this for the depot, where M is the Big-M notation which ensures that the constraint holds. Constraints (8)-(13) ensure that load capacity constraints hold. Time window constraints will be ensured by Constraint (14) - (17) and define variable τ. Constraints (18) - (20) define the variables z_{ij} and g_{ij} and that the capacity of the battery is not exceeded. Constraints (21) and (22) define the state of the charging of the battery. Finally, Constraint (23) ensures that the emission of the conventional vehicles is below the upper bound of emission and Constraint (24) makes sure that the domain of the decision variables.

3.1.2 The Green Vehicle Routing Problem with Steep Routes, Partial Battery Recharging and Time Windows

For the mathematical model, we will adjust the model that is covered in Macrina et al. (2019). First, we will modify the coefficient of energy consumption π used in Macrina et al. (2019) for the electric vehicles (EVs) in a way that it includes the road gradient and vehicle weight. Secondly,

Table 2: Estimation of emission factors for the G-VRP

Load of the vehicle	Weight laden (%)	Emission factor (kg CO_2/ km)
Empty	0	0.77
Low loaded	25	0.83
Half loaded	50	0.90
High loaded	75	0.95
Full load	100	1.01

we will modify the CO_2 emission $\epsilon(u_i^C$ for conventional vehicles (CVs) in a way that it takes road gradient into account. All other aspects in this model will be the same as formulated in Constraints (1) - (24).

3.2 Modelling CO_2 emission

For estimating the CO_2 emission, we need to have an emission factor ϵ that can calculate the emissions per kilometer. We assume that CO_2 emissions are only dependent on the type of vehicle and the quantity consumed by the vehicle. Furthermore, in the G-VRP we assume that the emission factor only depends on the mass of the vehicle and the load carried. For the G-VRPSR, the emission factor will also be dependent on the road gradient of the route. In order to calculate the emission factor we need to know the fuel conversion factor. Following Macrina et al. (2019), this factor will be 2.62 CO_2/ liter of diesel. Now, the estimated emission factor ϵ is equal to the consumption of diesel multiplied by the fuel conversion factor. Using the fact that the consumption of liter of diesel depends on the load of the vehicle for the G-VRP, we have summarized the emission factor for the G-VRP for different load in Table 2. In order to estimate the emission factor for the G-VRPSR, we need to include the road gradient in the consumption of a liter diesel. We will estimate this using the Comprehensive Modal Emission Model from Lai et al. (2021). The parameters used are defined in such a way that when the road gradient is zero, the emission factor will be the same as it would have been in the G-VRP. Furthermore, these parameters are summarized in Table 3 and are used for estimating the fuel consumption on an arc $(i, j) \in A$. The fuel consumption (FC) in liters of at arc $(i, j) \in A$ can be determined by formula 25

$$FC_{ij}(u_i) = \alpha_{ij}\frac{1}{v} + \beta_{ij}(w + u_i) + \gamma_{ij}v^2 \tag{25}$$

where

$$\alpha_{ij} = \xi\frac{1000FNVd_{ij}}{\kappa\psi}, \beta_{ij} = \xi\frac{d_{ij}(r + gsin\phi_{ij} + gC^{r)}cos\phi_{ij}}{\epsilon\omega\kappa\psi}, \gamma_{ij} = \xi\frac{0.5C^dA\rho d_{ij}}{\epsilon\omega\kappa\psi} \tag{26}$$

Table 3: Parameters for the Comprehensive Modal Emissions Model (CMEM) for estimating fuel consumption

Symbol	Description	Value
F	Engine friction factor ($kJ/rev/liter$)	0.13
N	Engine speed (rev/s)	30
V	Engine displacement (liters)	5
A	Frontal surface area of a vehicle (m^2)	5
C^d	Aerodynamic drag coefficients	0.35
C^r	Rolling resistance coefficients	0.005
r	Vehicle acceleration (m/s^2)	0
w	Curb weight (kg)	10000
κ	Heating value for diesel fuel (kJ/g)	42
ϵ	Vehicle drive train efficiency	0.3
ω	Efficiency parameter for diesel engines	0.6
ξ	Fuel-to-air mass ratio	1
ψ	Conversion factor from grams to liters	737
ρ	Air density (kg/m^3)	12041
g	Gravity (m/s^2)	9.81

Here, v is the vehicle speed, which we assume the same for all vehicles, for simplicity. u_i the payload (in kg) on a route and ϕ the road angle.

3.3 Modelling energy consumption

We model the energy consumption using Liu et al. (2017) for the G-VRPSR. In the G-VRP, we assume that the coefficient of energy consumption π is a constant factor and proportional to the distance travelled by the vehicle. However, for the G-VRPSR, we assume that the coefficient of energy consumption π is also proportional to the road gradient of the route. Liu et al. (2017) provided us with a regression formula of the energy consumption per km travelled, taking into account the distance of the route, the average speed of the route, whether airconditiong (A/C) is on or off, whether heater usage is on or off, whether the vehicle travels at night or at day and at last, it has for every road gradient a different dummy variable. We assume that A/C and heater usage are off and that the vehicle does not travel at night. Now, the road gradient is only dependent on the road gradient the distance of the road and the average speed. The regressions for different road gradients are summarized in Table 4.

4 Methodology

In this section, we will explain the methodologies that we will use. I will explain the algorithms used for the Green Vehicle Routing Problem in Section 4.1. Furthermore, I will explain the adjustments made in the algorithm of Section 4.1 for the Green Vehicle Routing Problem with

Table 4: Energy consumption per Km with road gradient

Road gradient	Energy consumption (per Km)
≤ -9%	0.040 - 0.003d_{ij} - 0.076v
-9% to -7%	0.155 - 0.003d_{ij} - 0.076v
-7% to -5%	0.224 - 0.003d_{ij} - 0.076v
-5% to -3%	0.251 - 0.003d_{ij} - 0.076v
-3% to -1%	0.299 - 0.003d_{ij} - 0.076v
-1% to 1%	0.372 - 0.003d_{ij} - 0.076v
1% to 3%	0.457 - 0.003d_{ij} - 0.076v
3% to 5%	0.524 - 0.003d_{ij} - 0.076v
5% to 7%	0.575 - 0.003d_{ij} - 0.076v
7% to 9%	0.678 - 0.003d_{ij} - 0.076v
9% to 11%	0.730 - 0.003d_{ij} - 0.076v
11% ≥	0.924 - 0.003d_{ij} - 0.076v

Steep Routes in Section 4.2.

4.1 The green vehicle routing problem

The proposed metaheuristic is based on the same iterated local search (ILS) used in Macrina et al. (2019). The algorithm used is summarized in Algorithm 1. Given the set of $\mathcal{N}$ customers that need to be served, we will first cluster the customers in two sets, the first set of customers are served by electrical vehicles (EVs) and the second set by conventional vehicles (CVs). After the clustering, we establish the initial routes for each vehicle. This will result in the initial solution. Then, we will apply local search and a perturbation on each iteration till the stopping criterion is met. Here, the stop criteria will be after 200 iterations. When the stopping criterion is satisfied, the best solution of all the solutions after an iteration is returned.

Algorithm 1 Iterated local search (ILS)

```
Generate the initial solution η0
Apply the local search procedure
while Stop criterion is not verified do
    Perturbation
    Local search
end while
return best solution η*
```

Constructing the initial solution In order to construct an initial solution, we first apply a clustering algorithm, then we will use insertion strategies for constructing feasible routes given the clusters for each type of vehicle. Let $\mathcal{S}$ be the set of all unserved customers. Furthermore, let $\mathcal{C}'$ be the set of all customers that will be served by a CV and $\mathcal{E}'$ the set of all customers served by an EV. That is, $\mathcal{S} = \mathcal{N} \setminus (\mathcal{C}' \cup \mathcal{E}')$. We initialize $\mathcal{C}'$ and $\mathcal{E}'$ by inserting the depot s in both sets. Then, for every iteration, till all customers are divided over the clusters, we will decide which customer is inserted in a cluster by the scores p_i^C and p_i^E, where the scores vary between 1 and 10. The score for the EVs p_i^E is calculated as follows:

$$p_i^E = 11 - \left(1 + \frac{d_i^E - d_{min}^E}{d_{max}^E - d_{min}^E} \cdot 9\right) \tag{27}$$

Here, d_i^E is the Euclidean distance from customer i to the barycentre of cluster $\mathcal{E}$, b_e, d_{min}^E the Euclidean distance of the nearest customer i, where $i \in \mathcal{S}$, to b_e and d_{max}^E the Euclidean distance of the furthest customer i, where $i \in \mathcal{S}$. The score for the is calculated as follows:

$$p_i^C = \lambda \left(11 - \left(1 + \frac{d_i^C - d_{min}^C}{d_{max}^C - d_{min}^C} \cdot 9\right)\right) + (1-\lambda)\left(1 + \frac{q_i^C - q_{min}^C}{q_{max}^C - q_{min}^C} \cdot 9\right) \tag{28}$$

where d_i^C is the Euclidean distance from customer i to the barycentre of cluster E b_c, d_{min}^C the Euclidean distance of the nearest customer i, where $i \in \mathcal{S}$, to b_c and d_{max}^C the distance of the farthest customer i, where $i \in \mathcal{S}$. Here, λ is set equal to 0.5. Furthermore, q_i^C is the demand of customer i, q_{min}^C the lowest customer demand of all served and unserved customers and q_{max}^C the largest customer demand of all served and unserved customers.

When all scores are calculated, we will assign the customer with the highest score in the corresponding cluster, so $i_E^* = \text{argmax}_{i \in \mathcal{S}}\{p_i^E\}$ and $i_C^* = \text{argmax}_{i \in \mathcal{S}}\{p_i^C\}$. If $i_E^* \neq i_C^*$, i_E^* will be assigned to cluster $\mathcal{E}'$, and i_C^* to $\mathcal{C}'$. Otherwise, if $p_{i^*}^E > p_{i^*}^C$, customer i^* will be assigned to cluster $\mathcal{E}'$, otherwise to cluster $\mathcal{C}'$. Finally, the depot s will be removed from both clusters.

Insertion strategy for conventional vehicles The aim of the insertion strategy is to construct feasible routes for conventional vehicles, by selecting the best unserved customer u^*, until the emission constraint is exceeded or till no customers are left in the cluster. If the emission constraint is exceeded but there are still customers unserved, the insertion strategy will stop and all unserved customers will be assigned to cluster $\mathcal{E}'$. We construct a route $(s, i_1, i_2, ..., i_m, t)$ by starting with an initial route (s, i, t), where s and t denote the depot and i_p the p-th customers in the route. The first customer in the initial route is the customer with the lowest closing time l_i. If the route is still feasible, this means that capacity constraints, emission constraints and time windows are still feasible, a new unserved customer $u^* \in \mathcal{C}'$ will be added to the route. The best customer u^* is chosen as follows. Calculate for every unserved customer the best position in the route by formula 29.

$$f_1(i(u), u, j(u)) = \min_{p \in \{1,...,m\}} c_{p-1,u} + c_{u,p} - c_{p-1,p} \tag{29}$$

where i(u) and j(u) are two adjacent customers in the current route. Finally, the customer will be decided by (30).

$$f_2(i(u^*), u^*, j(u^*)) = \max_u c_{s,u} - f_1(i(u), u, j(u)) \tag{30}$$

Before the insertion of u^*, the time window constraints, capacity constraints and emission constraints will be tested. If one or more of the time window constraints or capacity constraints are unsatisfied, u^* will not be inserted in the current route and a new route will be initialized. However, if the emission constraint is exceeded, u^* will also not be inserted, and all unserved customers will be served by an EV.

Insertion strategy for electrical vehicles The aim for this insertion strategy is to construct feasible routes for electrical vehicles, by selecting the best unserved customer u^* till no feasible solution is possible or till no customers are left in the cluster. We construct a route $\{s, i_1, i_2, ..., i_m, s\}$ with the same strategy used as with the conventional vehicles. However, before the insertion of the best customer u^*, only time window and capacity constraints are tested. When one or more constraints are exceeded before the insertion of the best customer u^*, energy capacity constraints of the route will be checked and recharge stations could possibly be added. If a recharge station should be added to the route, the recharge station that will be added is determined in the same manner as deciding the next customer in (30) and is not yet visited. Furthermore, time window constraints should be checked again and the route should be repaired if one or more time window constraints will not hold. We repair the route by iteratively removing the customer with the smallest time span $e_i - l_i$. In every iteration we also remove all recharge stations and check whether new recharge station should be added, after that we check time windows again, till the current route is feasible. If no feasible route can be constructed, all unserved customers will be assigned to a conventional vehicle. These new conventional routes will be constructed with the insertion strategy for conventional vehicles. However, the emission constraint can be violated. When the emission constraint is violated, we apply improvement heuristics with penalty function in the local search and perturbation.

Local search and perturbation In order to explore new feasible solutions, we introduce improvement heuristics based on the local search procedures. We distinguish between improvement heuristics for feasible solutions and improvement heuristics with penalty function for infeasible solutions. The improvement heuristic without penalty function has three different improvement strategies which are described as follows:

- **Change of nodes in conventional routes** For each conventional route, search for each customer the best feasible position in every other conventional route. The best position means the position with the largest cost reduction. The customer with the largest cost reduction will be relocated to his best other route.

- **Change of nodes in electrical routes** For each electrical route, search for each customer the best feasible position in every other electrical route. Here, the best position also means the position with the largest cost reduction, where new costs for possible new recharge station visits, removals of recharge stations and more battery charge are included. The customer with the largest cost reduction will be relocated to his best other electrical route. For both electrical routes, energy capacity constraints will be checked again and recharge stations could possibly be inserted/removed in the route. For deciding whether a recharge station should be removed, we remove all the recharge stations and allocate the current recharge stations till the battery capacity constraint is met.

- **Change of nodes in conventional and electrical routes** For each conventional and electrical route, search for the best feasible position in every other conventional or electrical route. If a relocation occurred where an electrical route is involved, the energy capacity constraints will be checked again and recharge stations could possibly be inserted/removed in the route.

For the improvement heuristic with penalty function, the same strategies as the improvement heuristic without penalty function will be used. However, the emission constraint is relaxed and the objective function will be defined as follows:

$$z'(\eta) = z(\eta) + \theta\epsilon(\eta) \tag{31}$$

where $z'(\eta)$ is the objective without a penalty function, θ the emission penalty and $\epsilon(\eta)$ the emission in the current solution. The value of θ is set equal to 1. After every iteration, this value will increase by 10% till the emission constraint is not longer violated. If the emission constraint is met, the improvement strategies without penalty function will be used.

4.2 The Green Vehicle Routing Problem with Steep Routes

For the Green Vehicle Routing Problem with Steep, the sequence of the route is a more important factor. This is because it is for example more efficient to drive down a vehicle with high capacity instead of driving upwards. For this purpose we add the Intra-route local search procedure, that was proposed in Brunner et al. (2021). We only apply this for routes carried out by conventional vehicles because energy consumption for electrical vehicles is not dependent on the load carried by the vehicle. This algorithm searches for improvements for a given route using three improvement strategies, where we only use two. Firstly, all possible two-arc exchanges within route r (2-opt), and secondly all possible swaps between two nodes in the route. The algorithm of the Iterated Local Search procedure for Steep Routes (ILSSR) is described in Algorithm 2.

Algorithm 2 Iterated Local Search for Steep Routes (ILSSR)

Generate the initial solution η_0
Apply the local search procedure
while Stop criterion is not verified **do**
    Perturbation
    Local search
    Inter-route local search
    Vehicle swap
end while
return best solution η^*

Here, the constructing of the initial solution will be done the same as in the ILS. However, the height difference between the customers will also be taken into account for calculating the distances between customers. Besides, the Local Search and Perturbation procedures are also the same as in the ILS. Furthermore, a vehicle swap improvement heuristic is added to the algorithm. This is a heuristic that swaps the electrical vehicle with a conventional vehicle for one route travelled by an electrical vehicle. The route that is chosen is the route that, when travelled by a conventional vehicle instead of an electrical vehicle, emits the least CO_2 in his route. Furthermore, the vehicle only switches if the new emission is below the upper bound of CO_2 emissions.

For comparing the costs of the routes of the G-VRP with the G-VRPSR, we will use the routes of the best solution that is generated from the ILS and recalculate the costs of the solution with new emissions and new energy consumption. If the new total emission exceeds the max emission, the conventional route with the lowest emissions will be converted to an electrical route and possible new recharge stations will be assigned. This continues till the total emission is below the max emission.

5 Results

We now initialize the results of the Iterated Local Search and the Iterated Local Search with Steep Routes. We carried out our tests on an Intel(R) Core(TM) I5-8250U CPU at 1.8 GHz having 8 GB of RAM using a Windows 10 operating system. The adjusted instances used in analysing the results are introduced in (Schneider et al., 2014) and are obtained from (Goeke, 2019). The instances are adjusted from the instances in (Solomon, 1987). The instances include for every customer and recharge station the coordinates, demand, opening and closing time and service time. The instances do not include the geographical height of the customers ad recharge stations, so we will generate those distances using a uniform random distribution. Furthermore, the instances are divided into three groups, C, RC and R and are different in their geographical distributions. Group C has a clustered distribution, R has a random distribution and RC has a combination of clustered and random distributions. Furthermore, 21 recharge stations are included in the instances. In the computational study, we first evaluate the Iterated Local Search (ILS) metaheuristic. Then we will compare the results obtained by ILS with the results obtained by the Iterated Local Search with the new improvement heuristics, where we assume zero height differences, to evaluate the proposed metaheuristic. After this, we compare the results obtained when we exclude steep routes with the results obtained when we include steep routes. We cover four different height differences. For altitudes differences between 0 and 10 meters, 0 and 50 meters, 0 and 100 meters and 0 and 250 meters.

5.1 Parameter setting

In order to analyse the results we need to clarify the parameters used in the ILS and ILSSR. The load capacity, battery capacity and refueling rate are stated in the instances. Here Macrina et al. (2019) used different values for these variables. Load capacity, in Macrina et al. (2019), was fixed at 500 kg, battery capacity at 20 KWh and the refueling rate was fixed at 20,000 kWh/h. However, we use for this the parameters given in Schneider et al. (2014), where the parameters differ per instance. The fuel consumption rate, only for the ILS, is equal to 1 and so is the velocity. The number of visits for recharge stations is 2, so σ equals 1 , because $|\mathcal{R}| = (\sigma + 1)\mathcal{R}$. For establishing the value of the upper bound (UB) on CO_2 emissions, we first need to calculate the value of the emissions in the worst case scenario UB_{max}, as in Macrina et al. (2019). The emission of the worst case is the emission that is emitted if every vehicle visited only one customer in the route and did not go back to the depot. Now, the parameter UB is set equal to $\alpha * UB_{max}$, where α is either 0.25, 0.50 or 0.75.

5.2 Analysing the ILS

Here, we investigate the generated initial solution with the ILS. As our ILS is very dependent on which improvement strategies are chosen in the first iterations, we will apply the ILS 20 times on every combination of the instance, α, and customer size. The reported solution is the best solution of the obtained 20 best solutions. The computation time is the average computation time of the 20 ILS metaheuristics.

To assess the performance of the ILS, we carried out a computational testing with the aim of comparing the quality of the solutions yielded by the proposed heuristic with the initial solution.

Specific results for the ILS for all customer sizes are shown in the Appendix, where we report, for each test instance and every value for α, the computation time, the costs of the initial solution and the cost of the best solution. The results are summarized in Table 5. Here the percentage gap in cost g_c is reported, where g_c is defined as $100 * g_c = -(c^B - c^I)/c^I$ and c^B is the cost that was generated from the ILS and c^I the cost of the initial solution. With averages varying between 30% and 59% for all values for α we clearly see that the ILS works and that we have obtained significantly better solutions. It is worth observing that some initial solutions were not feasible, so we did not consider those instances in the table.

Table 5: Summary results for ILS

		$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
		g_c	Run time (s)	g_c	Run time (s)	g_c	Run time (s)
$\|\mathcal{N}\| = 10$	Average	32%	0,3	30%	0,1	43%	0,09
	St. dev	21%		27%		25%	
	Minimum	2,3%		-50,9%		-9,7%	
	Maximum	66,2%		60,7%		80,8%	
$\|\mathcal{N}\| = 25$	Average	42%	2,2	54%	1,0	59%	0,7
	St. dev	17%		16%		14%	
	Minimum	14,6%		30,4%		31,3%	
	Maximum	73,2%		79,3%		80,3%	
$\|\mathcal{N}\| = 50$	Average	45%	10,8	53%	5,8	50%	5,1
	St. dev	20%		15%		29%	
	Minimum	1,3%		16,2%		-72,4%	
	Maximum	73,0%		73,6%		70,8%	
$\|\mathcal{N}\| = 100$	Average	52%	47,2	51%	26,2	52%	24,2
	St. dev	17%		19%		18%	
	Minimum	21,2%		0,0%		0,0%	
	Maximum	69,9%		71,0%		69,4%	

5.3 Investigating height differences in instances

The development of our algorithm requires, besides the parameters introduced in Section 5.1, the setting of the energy consumption and fuel consumption. First, we investigate the energy consumption per kilometer travelled in kWh. Summary statistics are given in Table 6 for areas with different maximum heights H, where customers can be located between 0 and 10 meter, 0 and 50 meter, 0 and 100 meter, and 0 and 250 meter. The table shows that for all different height ranges between 0 and H the average energy consumption is around 0.293, however the standard deviation, the minimum value and the maximum value do change when the height ranges become higher. Because the average energy consumption is 0.293 per kilometer, and we assumed an energy consumption of 1.0 per kilometer in the ILS, we have multiplied the energy consumption in the ILSSR by 1.0/0.293.

Table 6: Summary statistics for energy consumption per kilometer

	H = 10	H = 50	H = 100	H = 250
Mean	0.293	0.2932	0.293	0.293
St. dev	0	0.003	0.009	0.022
Minimum	0.219	0.145	-0.039	-0.039
Maximum	0.378	4.030	0.845	0.845

For investigating the CO_2 emissions, the summary statistics of the CO_2 emission per kilometer are shown in Table 7 for different height ranges H. It clearly shows more variation in CO_2 emissions between customers when the height range increases.

Table 7: Summary statistics for CO_2 emission per kilometer

	H = 10	H = 50	H = 100	H = 250
Mean	0.837	0.841	0.853	0.922
St. dev	0.042	0.189	0.357	0.828
Minimum	0.005	0.002	0.002	0.002
Maximum	2.777	10.519	20.102	47.381

5.4 Analysing ILSSR in flat areas

To asses the performance of the ILSSR, we first investigate whether the metaheuristic obtains better results than the ILS in flat areas. And so, we investigate whether the implementation of the inter route search and the route swap improvement heuristic lead to better results. Summary statistics are given in Table 8, where the run time percentage gap t_c is defined as $100 * t_c = (r^A - r^B)/r^B$ and r^B is the run time for the ILS and r^A the run time for the ILSSR. Furthermore, the cost percentage gap g_c is defined as $100 * g_c = -(c^A - c^B)/c^B$ and c^B is the cost that was generated from the ILS and c^A the cost obtained from the ILSSR. Specific results for the ILSSR without steep routes are shown in the Appendix. Looking at Table 8, it is evident that when we use the ILSSR cost will decrease with on average 10%. However, in some cases we still observe a cost increase for the ILSSR. The computational results clearly show an advantage for the ILSSR in flat areas in terms of efficiency. This advantage becomes more evident for instances with 50 customers, where the ILSSR is on average 17% faster than the ILS. However, for smaller instances the ILSSR is slower.

5.5 Analysing the ILSSR with height differences

In this section, we investigate the impact of height differences on the obtained results. We compare the obtained results for different height ranges with both the ILS and the ILSSR without height differences, because the cost improvement or increment can also be caused by

Table 8: Summary statistics for comparing ILSSR with ILS in flat areas

		$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
		g_c	t_c	g_c	t_c	g_c	t_c
$\lvert\mathcal{N}\rvert = 10$	Average	9,4%	18,3%	9,5%	20,1%	18,1%	36,6%
	St. dev	11%		18%		21%	
	Minimum	1%		-7%		-3%	
	Maximum	40%		75%		64%	
$\lvert\mathcal{N}\rvert = 25$	Average	7,2%	7,3%	12,5%	10,3%	2,4%	40,6%
	St. dev	9%		12%		16%	
	Minimum	-10%		-6%		-40%	
	Maximum	26%		31%		43%	
$\lvert\mathcal{N}\rvert = 50$	Average	11,1%	-2,4%	13,2%	-18,8%	14,9%	-31,8%
	St. dev	21%		13%		13%	
	Minimum	-7%		-20%		-18%	
	Maximum	98%		28%		39%	

the implementation of the inter route search or the vehicle swap improvement heuristic.

5.5.1 Comparing with ILS

For comparing the results of the ILS, we focus on three different height ranges, an area with height differences between 0 and 10 meters, 0 and 50 meters, and 0 and 100 meters. We use the same instances for every different height range, however only the height where the customers and recharge stations are located are different. We compare the solutions generated by the ILSSR, where road gradient were taken into account, with the solutions generated by the ILS, were we neglect the road gradient. As mentioned before, the costs of the best solution of the ILS were recalculated with new battery use for electrical routes, and were conventional routes could be assigned as electrical routes if the new emission exceeds the maximal emission. We have summarized the percentage cost gain g_c of every instance and for every height difference H in Tables 9 - 11, where $g_c = 100*(c^S - c^B)/c^B$ and c^S is the cost generated by the ILSSR and c^B the adjusted cost generated by the ILS. It is worth observing that for every height range the average cost gain is positive. So the ILSSR obtains better results for all height differences. Furthermore, especially for the results when the height range is set equal to 10 meters, the average run time is also faster for the ILSSR. However, it is worth observing that the cost gain can be dependent from the implementation of the inter route local search and vehicle swap improvement heuristic rather than taking steep routes into account for defining the energy consumption and CO_2 emission. Furthermore, it is also worth observing that the ILSSR does not obtains the best solution for every instance.

Table 9: Summary statistics for comparing ILSSR with ILS with H = 10

		α = 0.25		α = 0.50		α = 0.75	
		g_c	t_c	g_c	t_c	g_c	t_c
$\lvert\mathcal{N}\rvert = 10$	Average	9,2%	-9,1%	4,0%	-17,2%	17,6%	4,9%
	St. dev	17%		7%		17%	
	Minimum	-14%		-12%		-6%	
	Maximum	53%		18%		48%	
$\lvert\mathcal{N}\rvert = 25$	Average	2,0%	-15,1%	12,6%	-9,3%	4,5%	6,2%
	St. dev	8%		12%		12%	
	Minimum	-8%		-6%		-7%	
	Maximum	23%		30%		41%	
$\lvert\mathcal{N}\rvert = 50$	Average	3,8%	-14,9%	13,0%	-27,5%	12,8%	-41,6%
	St. dev	8%		13%		12%	
	Minimum	-15%		-28%		-13%	
	Maximum	18%		29%		27%	

Table 10: Summary statistics for comparing ILSSR with ILS with H = 50

		α = 0.25		α = 0.50		α = 0.75	
		g_c	t_c	g_c	t_c	g_c	t_c
$\lvert\mathcal{N}\rvert = 10$	Average	12,8%	11,1%	4,0%	28,5%	17,5%	18,8%
	St. dev	17%		7%		17%	
	Minimum	-7%		-12%		-2%	
	Maximum	53%		18%		48%	
$\lvert\mathcal{N}\rvert = 25$	Average	1,8%	6,0%	11,4%	2,3%	4,3%	72,9%
	St. dev	10%		11%		12%	
	Minimum	-25%		-6%		-15%	
	Maximum	23%		30%		34%	
$\lvert\mathcal{N}\rvert = 50$	Average	3,4%	-6,6%	12,2%	-16,4%	13,0%	-33,8%
	St. dev	8%		13%		12%	
	Minimum	-13%		-23%		-19%	
	Maximum	16%		31%		27%	

Table 11: Summary statistics for comparing ILSSR with ILS with H = 100

		α = 0.25		α = 0.50		α = 0.75	
		g_c	t_c	g_c	t_c	g_c	t_c
$\lvert\mathcal{N}\rvert = 10$	Average	11,5%	10,8%	7,3%	-7,9%	18,1%	50,1%
	St. dev	19%		15%		18%	
	Minimum	-27%		-12%		-6%	
	Maximum	53%		62%		64%	
$\lvert\mathcal{N}\rvert = 25$	Average	1,4%	6,8%	7,0%	27,6%	3,7%	37,9%
	St. dev	5%		10%		12%	
	Minimum	-7%		-9%		-22%	
	Maximum	12%		30%		33%	
$\lvert\mathcal{N}\rvert = 50$	Average	3,7%	0,8%	12,0%	-5,2%	18,0%	-34,6%
	St. dev	8%		12%		21%	
	Minimum	-13%		-12%		-12%	
	Maximum	25%		30%		100%	

5.5.2 Comparing with ILSSR without height differences

We finally evaluate the results obtained by the ILSSR for different height ranges H, an area with height differences between 0 and 10 meters, 0 and 50 meters, 0 and 100 meters, and 0 and 250 meters. We use the same instances for every different height range, however only the height where the customers and recharge stations are located are different. We compare the ILSSR where we set the height range on a specific height with the ILSSR where we assumed no height differences when traveling between customers. The reported values of the ILSSR where we set the height range equal to zero, are the adjusted results described in Section 4.2. Table 12 presents the cost percentage gap g_c defined as $100 * g_c = -(c^A - c^B)/c^B$, for every customer size $\mathcal{N}$ and height range H, where c^C is the cost that was generated from the ILSSR including the height of every customer and c^A the cost obtained from the ILSSR where we assumed flat areas. Furthermore, r_c is defined as the percentage of instances where including height differences between customers lead to a cost decrease. If r_c is below 50%, the excluding height differences between customers in the ILSSR obtains better results. The values for r_c clearly demonstrates that around a height range of 100, but especially from a height range of 250, including height differences between customers for defining CO_2 emission and energy consumption obtains better results.

Table 12: Average cost decrease for including steep routes

		$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
		g_c	r_c	g_c	r_c	g_c	r_c
H = 10	$\lvert\mathcal{N}\rvert = 10$	-1,8%	42%	-0,2%	83%	-2,5%	75%
	$\lvert\mathcal{N}\rvert = 25$	-3,0%	29%	0,3%	54%	-1,3%	38%
	$\lvert\mathcal{N}\rvert = 50$	-0,5%	33%	-0,1%	42%	0,6%	58%
H = 50	$\lvert\mathcal{N}\rvert = 10$	2,8%	50%	-0,2%	83%	-2,6%	71%
	$\lvert\mathcal{N}\rvert = 25$	-0,9%	42%	-1,0%	33%	-1,1%	50%
	$\lvert\mathcal{N}\rvert = 50$	-0,1%	50%	-0,6%	38%	0,8%	58%
H = 100	$\lvert\mathcal{N}\rvert = 10$	2,9%	50%	2,9%	83%	-6,3%	71%
	$\lvert\mathcal{N}\rvert = 25$	-0,1%	46%	1,4%	38%	-1,6%	38%
	$\lvert\mathcal{N}\rvert = 50$	1,4%	54%	2,4%	63%	1.4%	58%
H = 250	$\lvert\mathcal{N}\rvert = 10$	1,6%	58%	6,7%	79%	-2,4%	67%
	$\lvert\mathcal{N}\rvert = 25$	3,9%	75%	11,6%	75%	2,2%	54%
	$\lvert\mathcal{N}\rvert = 50$	4,4%	79%	10,4%	92%	3,0%	71%

The cost percentage gap for $\mathcal{N}$ is shown in Figure 1 for every value of α. It clearly demonstrates an increase in the cost percentage gap when the height differences between customers becomes larger.

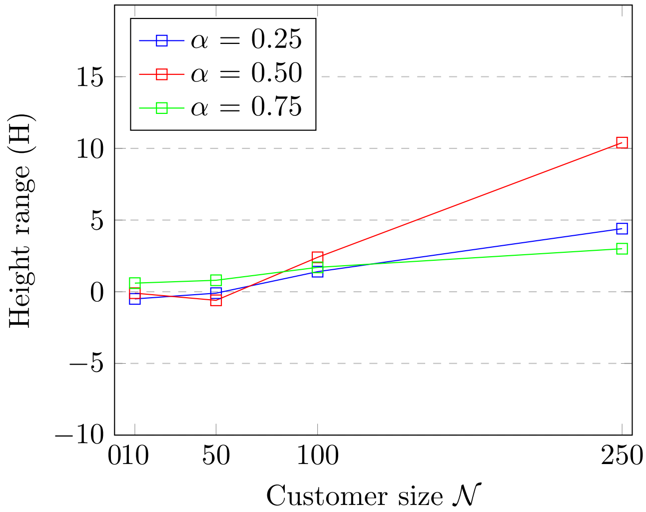

Figure 1: Average cost decrease for including steep routes

6 Conclusions

In this thesis, we have introduced the Green Vehicle Routing Problem with partial battery recharging, time windows and a mixed fleet of conventional vehicles and electrical vehicles with and without steep routes. We first analysed the performances of the iterated local search where we neglect the road gradients. Secondly, we proposed an iterated local search metaheuristic where we include the road gradients to estimate the CO_2 emissions and energy consumption more precise. Furthermore, we implemented an inter route search and a vehicle swap improvement heuristic. We tested this for four height differences, between 0 and 10 meters, 0 and 50 meters, 0 and 100 meters, and 0 and 250 meters, to analyse at what level of height differences it is important to include the road gradient. Our test results have shown on one hand that when we compare the proposed metaheuristic with the original metaheuristic, that for all height differences we can decrease costs by including the road gradients in the estimation of CO_2 emission and energy consumption. However, on the other hand when we compare the proposed metaheuristic where we set the height differences to the real height differences with the same proposed metaheuristic where we set the height differences equal to zero, we see that we only obtain a cost decrease from 100 meters. Furthermore, the proposed metaheuristic shows that finding solutions does not take much longer to generate. This entails that the ILSSR metaheuristic is efficient and recommended to use, especially when height differences between locations increase.

For future research, it could be interesting to investigate whether for other vehicles with other characteristics, the road gradient should be included for lower height differences or for higher height differences. So, to research what impact the vehicle has for deciding when to include road gradients. Furthermore, it is interesting to investigate whether varying recharging costs for different recharge stations are important to include and to investigate how the proposed metaheuristic behaves when we have multiple depots.

References

Agency, I. E. (2018). Transport sector co2 emissions by mode in the sustainable development scenario, 2000-2030. *iea.org*.

Bandeira, J., Almeida, T. G., Khattak, A. J., Rouphail, N. M. & Coelho, M. C. (2013). Generating emissions information for route selection: Experimental monitoring and routes characterization. *Journal of Intelligent Transportation Systems*, *17*(1), 3-17.

Bektaş, T. & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, *45*(8), 1232-1250. (Supply chain disruption and risk management)

Brunner, C., Giesen, R., Klapp, M. A. & Flórez-Calderón, L. (2021). Vehicle routing problem with steep roads. *Transportation Research Part A: Policy and Practice*, *151*, 1-17.

Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91.

Demir, E., Bektaş, T. & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346-359.

Demir, E., Bektaş, T. & Laporte, G. (2014). The bi-objective pollution-routing problem. *European Journal of Operational Research*, *232*(3), 464-478.

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T. & Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, *56*, 265-293.

Goeke, D. (2019). E-vrptw instances. *Mendeley Data*. Retrieved from `doi:10.17632/h3mrm5dhxw.1`

Jabali, O., Van Woensel, T. & de Kok, A. (2012). Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, *21*(6), 1060-1074.

Lai, D., Costa, Y., Demir, E., Florio, A. & Van Woensel, T. (2021, 05). The pollution-routing problem with speed optimization and uneven topography.

Lin, J., Zhou, W. & Wolfson, O. (2016). Electric vehicle routing problem. *Transportation Research Procedia*, *12*, 508-521. (Tenth International Conference on City Logistics 17-19 June 2015, Tenerife, Spain)

Liu, Yamamoto, T. & Morikawa, T. (2017). Impact of road gradient on energy consumption of electric vehicles. *Transportation Research Part D: Transport and Environment*, *54*, 74-81.

Liu, L. & Lai, L. (2021, 06). An effective heuristic for multidepot low-carbon vehicle routing problem. *Mathematical Problems in Engineering*, *2021*, 1-10.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F. & Laporte, G. (2019). The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Computers Operations Research*, *101*, 183-199.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, *23*(5), 377-386.

Montoya, A., Guéret, C., Mendoza, J. E. & Villegas, J. G. (2016). A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, *70*, 113-128.

Palmer, A. (2007). The development of an integrated routing and carbon dioxide emissions model for goods vehicles.

Sassi, O., Ramdane Cherif-Khettaf, W. & Oulamara, A. (2014, 10). Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs.

Schneider, M., Stenger, A. & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*, 500-520.

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

Suzuki, Y. (2011). A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, *16*(1), 73-77.

Tajik, N., Tavakkoli-Moghaddam, R., Vahdani, B. & Meysam Mousavi, S. (2014). A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty. *Journal of Manufacturing Systems*, *33*(2), 277-286.

Yu, V. F., Jodiawan, P. & Gunawan, A. (2021). An adaptive large neighborhood search for the green mixed fleet vehicle routing problem with realistic energy consumption and partial recharges. *Applied Soft Computing*, *105*, 107251.

Zhang, J., Zhao, Y., Xue, W. & Li, J. (2015). Vehicle routing problem with fuel consumption and carbon emission. *International Journal of Production Economics*, *170*, 234-242.

A Results for ILS

Table 13: Results for ILS for the instances with $|\mathcal{N}| = 10$

	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$		
Instance	Time(ms)	Cost IS	Cost	Time(ms)	Cost IS	Cost	Time(ms)	Cost IS	Cost
C101C10	139	516	249	91	429	169	65	466	89.5
C102	47	509	172	144	429	169	98	359	91
C103	51	401	169	50	430	169	38	360	169
C104	41	297 *	145**	43	297	168	42	297	168
C105	47	414	174	54	329	168	42	330	88.8
C106	49	402	281	50	330	168	62	330	88.8
C107	49	414	174	52	330	168	44	330	88.8
C108	38	402	169	54	330	169	57	330	88.8
R101	454	653	591	209	591	540	165	528	451
R102	555	581	521	222	519	451	116	519	436
R103	502	265*	476	206	265*	400	121	288	316
R104	1221	433	460	122	433	366	89	433	333
R105	418	588	541	291	398	462	196	398	384
R106	330	583	512	177	401	423	134	401	324
R107	1048	449	459	337	449	365	220	472	284
R108	315	434	424	92	434	325	58	434	282
RC101	270	582	412	107	657	318	74	657	318
RC102	335	398	398	74	398	287	66	398	287
RC103	171	398	283	49	398	284	62	398	284
RC104	161	407	282	47	407	288	53	407	288
RC105	145	457	410	75	550	250	66	550	216
RC106	161	439	410	73	500	319	73	500	177
RC107	305	472	296	180	472	298	234	472	212
RC108	344	398	278	107	398	287	75	398	284

IS = initial solution

* = initial solution is not feasible

Table 14: Results for ILS for the instances with $|\mathcal{N}| = 25$

	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$		
Instance	Time(s)	Cost IS	Cost	Time(s)	Cost IS	Cost	Time(s)	Cost IS	Cost
C101	2.6	1268	443	0.6	1222	316	0.5	1047	238
C102	1.0	1407	377	0.5	1240	303	0.5	1108	224
C103	0.8	1157	375	0.4	1020	302	0.5	1085	214
C104	0.8	655	357	0.5	796	208	0.1	437	197
C105	1.7	1255	456	0.8	1222	316	0.5	1062	238
C106	1.3	961	415	0.5	939	233	0.5	915	227
C107	1.7	1242	452	0.5	1209	250	0.5	959	247
C108	1.7	959	419	0.6	877	314	0.1	570	169
R101	2.3	1462	1062	1.3	1297	903	0.9	1165	730
R102	3.0	1282	964	1.3	1158	781	0.6	1086	567
R103	2.6	1230	858	1.5	962	619	0.7	960	433
R104	4.1	1001	767	1.6	1052	612	1.0	1052	414
R105	6.0	1338	948	2.4	1259	740	1.3	1165	563
R106	2.7	1336	852	1.4	1058	638	0.7	1101	511
R107	4.6	1178	740	2.2	1144	509	1.1	916	385
R108	2.4	1066	707	1.3	973	500	0.5	1130	385
RC101	1.5	1559	911	0.8	1424	702	0.8	1289	702
RC102	1.3	1255	695	0.8	959	568	0.7	959	412
RC103	1.7	1007	554	0.9	997	428	0.9	997	376
RC104	1.4	922	566	1.0	772	321	1.0	772	319
RC105	1.4	1381	914	0.8	1043	717	0.8	1043	717
RC106	1.5	1126	769	0.8	873	576	0.7	873	581
RC107	2.2	678	579	1.3	678	344	1.3	678	353
RC108	1.6	678	568	0.7	678	324	0.6	678	300

IS = initial solution

* = initial solution is not feasible

Table 15: Results for ILS for the instances with $|\mathcal{N}| = 50$

	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$		
Instance	Time(s)	Cost IS	**Cost**	Time(s)	Cost IS	**Cost**	Time(s)	Cost IS	Cost
C101	10.58	2622	753	5.98	2286	604	6.83	2047	622
C102	13.58	2135	712	6.12	1645	612	4.60	1645	604
C103	10.63	1673	718	4.18	1342	622	3.91	1342	596
C104	6.53	1441	684	5.06	1275	595	4.87	1275	580
C105	8.3	2529	684	3.46	1903	604	3.46	2062	603
C106	9.29	2325	714	4.07	1829	605	4.69	1829	608
C107	9.11	2512	683	4.45	2182	604	3.73	2022	603
C108	5.12	2049	687	4.19	1698	649	5.12	1633	650
R101	11.2	1748*	1928	6.20	1804	1511	3.25	1801	1187
R102	11.51	1641*	1691	5.25	1665	1226	3.28	971	1674
R103	13.51	2553	1302	8.78	2280	965	4.41	2196	859
R104	12.7	2102	1268	7.74	1823	1010	8.24	1823	927
R105	22.71	1628	1607	10.65	1639	1210	6.10	1627	893
R106	12.28	1568	1451	5.12	1559	1052	4.00	1533	899
R107	25.03	1925	1196	13.85	1775	821	11.28	1775	796
R108	12.42	1903	1074	7.64	1782	737	6.64	1782	683
RC101	7.58	2509	1643	3.95	3034	1175	4.37	3034	1176
RC102	8.55	2294	1413	3.58	2400	1121	3.05	2400	1005
RC103	6.94	1750	1239	3.55	1960	730	3.21	1960	726
RC104	8.04	2007	1170	4.39	1848	570	5.03	1848	621
RC105	9.14	2805	1539	4.35	2356	1176	3.86	2356	1203
RC106	5.99	2420	1603	4.23	2462	1113	4.69	2462	1316
RC107	14.54	1485	1192	7.62	1485	877	9.49	1185	931
RC108	5.87	1755	999	4.25	1638	901	4.05	1638	898

IS = initial solution

* = initial solution is not feasible

Table 16: Results for ILS for the instances with $|\mathcal{N}| = 100$

	a = 0.25			**a= 0.50**			**a = 0.75**		
Instance	**Time(s)**	**Cost IS**	**Cost**	**Time(s)**	**Cost IS**	**Cost**	**Time(s)**	**Cost IS**	**Cost**
C101	44.5	5509	1723	28.8	4933	1431	28.04	4839	1483
C102	45.63	5121	1603	32.1	4026	1439	29.4	4026	1436
C103	35.5	4195	1594	28.6	3480	1435	29.2	3480	1514
C104	33.2	3516	1565	22.9	2851	1181	21.2	2851	1309
C105	46.2	5563	1672	26.3	4743	1503	27.4	4707	1452
C106	41.2	4737	1679	30.9	4399	1375	31.2	4410	1379
C107	42.9	5404	1747	25.9	4550	1397	25.4	4513	1410
C108	43.8	4268	1709	27.9	3960	1491	30.1	3960	1427
R101	50.4	3112*	2913	23.9	3105	2122	15.3	3131	1643
R102	52.4	2746*	2505	19.9	2876	1695	16.4	2862	1578
R103	40.9	2687*	2091	31.2	3621	1647	27.3	3621	1571
R104	61.5	3238	1734	43.7	2737	1407	36.7	2737	1409
R105	62.8	2611*	2261	21.8	3206	1634	20.5	3213	1528
R106	62.3	2905*	2156	20.9	2957*	1395	35.5	3087	1376
R107	72.2	3592	1871	51.5	3198	1286	32.1	3198	1367
R108	64.7	1819*	1593	15.4	1934	1045	15.3	1934	1052
RC101	47.3	3851	2941	17.4	3726	1922	18.9	3726	1846
RC102	46.4	3195	2517	21.4	3751	1795	19.0	3751	1726
RC103	42.8	3672*	2199	21.4	3698	1590	21.9	3698	1464
RC104	49.0	2524*	1834	17.8	2559	1343	14.7	2559	1271
RC105	44.4	3596*	2445	20.2	3631	1674	16.3	3631	1521
RC106	51.3	3606	2298	23.6	3312	1436	18.9	3312	1495
RC107	47.9	3046*	1975	21.3	3377	3377	21.6	3377	3377
RC108	49.2	2839	1824	27.9	3346	3346	28.7	3346	3346

IS = initial solution

* = initial solution is not feasible

B Results for ILSSR with flat areas

Table 17: Results for ILSSR without height differences for the instances with $|\mathcal{N}| = 10$

	Run time	Cost	Run time	Cost	Run time	Cost
C101	188	171	74	169	88	89
C102	166	171	102	169	110	89
C103	228	168	111	168	184	89
C104	57	141	34	168	57	61
C105	50	168	39	168	52	88
C106	42	168	39	168	54	88
C107	51	168	67	168	70	88
C108	45	168	61	168	58	88
R101	449	557	486	267	428	179
R102	555	474	285	424	179	352
R103	426	678	550	375	230	325
R104	447	399	108	354	122	287
R105	277	475	218	401	172	384
R106	455	464	205	389	102	313
R107	912	398	352	353	243	285
R108	565	400	155	348	158	280
RC101	257	395	96	317	70	237
RC102	157	391	78	281	61	201
RC103	186	267	43	258	56	169
RC104	152	267	51	261	63	169
RC105	159	392	249	63	54	169
RC106	148	273	41	260	46	169
RC107	150	280	67	249	51	169
RC108	166	262	42	248	66	166

Table 18: Results for ILSSR without height differences for the instances with $|\mathcal{N}| = 25$

	Run time	Cost	Run time	Cost	Run time	Cost
C101	0,9	438	0,6	237	0,4	237,0
C102	2,1	415	1,0	224	1,0	226,0
C103	2,3	375	1,4	216	1,5	219,0
C104	0,7	283	0,4	194	0,5	212,0
C105	1,2	429	0,6	235	0,5	235,0
C106	1,4	410	0,5	227	0,4	228,0
C107	1,2	432	0,6	238	0,5	235,0
C108	1,6	408	0,7	232	0,6	236,0
R101	2,6	1023	1,7	885	1,3	744,0
R102	3,3	889	2,0	763	1,3	578,0
R103	3,1	797	1,8	618	0,8	498,0
R104	2,8	733	1,2	571	0,6	403,0
R105	3	891	1,6	721	0,8	580,0
R106	3,3	811	2,1	618	0,9	514,0
R107	4,7	682	2,0	489	0,8	385,0
R108	5,1	667	2,2	489	0,8	385,0
RC101	1,1	842	0,6	604	0,4	478,0
RC102	1,4	707	0,5	409	0,4	385,0
RC103	1,5	418	0,5	309	0,5	317,0
RC104	1,5	418	0,5	305	0,5	313,0
RC105	1,2	837	0,7	531	0,4	410,0
RC106	1,6	731	0,8	397	0,5	396,0
RC107	1,3	558	0,6	366	0,5	372,0
RC108	1,4	424	0,5	307	0,5	308,0

Table 19: Results for ILSSR without height differences for the instances with $|\mathcal{N}| = 50$

	Run time	Cost	Run time	Cost	Run time	Cost
C101	9,5	622	2,7	462	2,5	466,0
C102	19,4	621	7,9	443	7,7	461,0
C103	17	729	2,9	464	3,1	483,0
C104	5,7	568	2,8	435	2,6	422,0
C105	7,6	622	3,3	449	2,6	453,0
C106	6,3	721	2,8	462	2,9	469,0
C107	7,6	621	3,4	464	3,4	463,0
C108	9,1	627	3,2	488	2,9	487,0
R101	13,6	1710	8,5	1441	4,3	1171,0
R102	13,1	1611	10,8	1239	3,8	1013,0
R103	14,4	1259	6,9	925	3,1	788,0
R104	10,5	1214	5,1	805	2,7	708,0
R105	13,2	1542	8,9	1199	4,0	895,0
R106	14,8	1301	7,3	1046	3,4	806,0
R107	17	1137	7,7	794	2,9	723,0
R108	17,7	17,7	6,5	739	3,6	680,0
RC101	7,1	1707	2,7	1049	2,1	973,0
RC102	6,8	1405	2,7	1021	2,3	961,0
RC103	6	1159	2,6	717	2,4	774,0
RC104	6,6	788	2,6	686	2,4	733,0
RC105	6,9	1367	2,5	875	2,3	937,0
RC106	6,3	1434	2,5	929	2,3	934,0
RC107	7,2	1230	2,4	721	2,4	805,0
RC108	5,7	1067	2,3	707	2,2	787,0

C Results for ILSSR with height differences

Table 20: Results for ILSSR with H = 10 for the instances with $\mathcal{N}$ = 10

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	171	159.0	169	79.0	89	76.0
C102	171	54.0	169	40.0	89	62.0
C103	168	54.0	168	38.0	89	58.0
C104	145	39.0	168	29.0	88	53.0
C105	168	39.0	168	37.0	88	51.0
C106	168	39.0	168	39.0	88	54.0
C107	168	63.0	168	44.0	88	54.0
C108	168	52.0	168	43.0	88	67.0
R101	557	396.0	486	238.0	428	155.0
R102	493	381.0	425	221.0	367	145.0
R103	426	441.0	375	210.0	323	139.0
R104	399	434.0	354	109.0	287	119.0
R105	475	480.0	401	366.0	384	206.0
R106	464	304.0	389	175.0	327	102.0
R107	398	434.0	353	165.0	286	121.0
R108	400	500.0	348	115.0	280	167.0
RC101	396	269.0	317	122.0	237	68.0
RC102	391	145.0	281	76.0	201	59.0
RC103	267	175.0	258	43.0	169	53.0
RC104	267	139.0	261	46.0	169	57.0
RC105	392	166.0	249	59.0	169	55.0
RC106	273	152.0	260	41.0	169	45.0
RC107	280	139.0	249	61.0	169	53.0
RC108	262	175.0	248	47.0	166	58.0

Table 21: Results for ILSSR with H = 50 for the instances with $\mathcal{N}$ = 10

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	171	0.1	169	0.0	89	0.0
C102	171	0.0	169	0.0	89	0.0
C103	168	0.0	168	0.0	89	0.0
C104	141	0.0	168	0.0	88	0.0
C105	168	0.0	168	0.0	88	0.0
C106	169	0.0	168	0.0	88	0.0
C107	171	0.0	168	0.0	88	0.0
C108	168	0.0	168	0.0	88	0.0
R101	557	0.4	486	0.2	428	0.1
R102	475	0.3	414	0.1	381	0.1
R103	434	1.8	375	0.6	325	0.3
R104	399	1.5	354	0.4	287	0.3
R105	475	0.9	401	0.6	384	0.5
R106	437	0.4	390	0.3	313	0.1
R107	398	0.3	353	0.1	286	0.1
R108	400	0.4	348	0.1	280	0.1
RC101	395	0.2	317	0.0	237	0.0
RC102	391	0.1	281	0.0	201	0.0
RC103	267	0.1	258	0.0	169	0.0
RC104	267	0.1	261	0.0	169	0.0
RC105	392	0.1	253	0.0	173	0.0
RC106	273	0.1	260	0.0	169	0.0
RC107	280	0.1	249	0.0	169	0.0
RC108	262	0.4	248	0.3	166	0.1

Table 22: Results for ILSSR with H = 100 for the instances with $\mathcal{N}$ = 10

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	171	271.0	169	134.0	89	123.0
C102	171	173	169	125	89	168
C103	168	79.0	168	49.0	89	89.0
C104	140	35.0	64	60.0	61	302.0
C105	171	49.0	168	52.0	88	108.0
C106	171	61.0	168	39.0	89	68.0
C107	172	66.0	168	42.0	88	61.0
C108	168	54.0	168	35.0	88	50.0
R101	556	505.0	488	239.0	428	156.0
R102	494	314.0	424	190.0	367	139.0
R103	426	679.0	376	239.0	323	176.0
R104	399	468.0	354	122.0	287	163.0
R105	475	257.0	401	204.0	384	157.0
R106	464	290.0	390	166.0	327	118.0
R107	398	404.0	353	143.0	285	115.0
R108	400	455.0	348	106.0	280	183.0
RC101	395	314.0	317	127.0	237	66.0
RC102	391	149.0	281	78.0	201	60.0
RC103	333	194.0	258	46.0	169	49.0
RC104	268	137.0	261	51.0	172	53.0
RC105	392	149.0	249	83.0	169	52.0
RC106	274	149.0	254	41.0	169	48.0
RC107	266	199.0	249	83.0	169	57.0
RC108	262	194.0	248	45.0	169	67.0

Table 23: Results for ILSSR with H = 250 for the instances with $\mathcal{N} = 10$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	248	141.0	169	106.0	169	68.0
C102	248	61.0	169	43.0	169	38.0
C103	170	57.0	168	43.0	168	35.0
C104	141	40.0	141	37.0	62	53.0
C105	248	61.0	168	**43.0**	168	39.0
C106	248	60.0	171	43.0	168	42.0
C107	248	60.0	168	37.0	168	39.0
C108	248	76.0	170	37.0	168	32.0
R101	585	837.0	558	496.0	486	544.0
R102	473	384.0	445	208.0	367	149.0
R103	468	346.0	385	288.0	340	143.0
R104	404	432.0	354	148.0	354	99.0
R105	475	276.0	400	222.0	284	169.0
R106	472	320.0	410	239.0	368	148.0
R107	395	616.0	362	202.0	278	126.0
R108	397	488.0	376	254.0	312	200.0
RC101	396	277.0	318	163.0	317	83.0
RC102	391	144.0	281	86.0	312	65.0
RC103	266	195.0	255	47.0	172	59.0
RC104	262	154.0	256	55.0	174	61.0
RC105	392	158.0	312	103.0	174	74.0
RC106	275	144.0	256	49.0	260	47.0
RC107	283	150.0	253	64.0	176	64.0
RC108	245	176.0	250	46.0	166	60.0

Table 24: Results for ILSSR with H = 10 for the instances with $\mathcal{N} = 25$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	438	1.0	238	0.6	237	0.4
C102	320	1.0	217	0.5	227	0.4
C103	406	0.9	216	0.5	230	0.5
C104	353	0.7	195	0.4	211	0.4
C105	429	1.0	235	0.8	254	0.5
C106	417	1.2	227	0.5	229	0.4
C107	430	1.0	235	0.6	237	0.4
C108	411	1.3	226	0.5	242	0.4
R101	1021	2.4	881	1.5	746	1.0
R102	875	2.6	706	1.7	579	1.1
R103	781	2.7	614	1.5	499	0.6
R104	708	2.5	580	1.1	403	0.6
R105	891	2.8	724	1.5	575	0.8
R106	811	2.7	609	1.7	517	0.8
R107	733	3.2	489	1.5	385	0.6
R108	663	3.9	489	1.5	390	0.5
RC101	842	1.0	604	0.6	523	0.4
RC102	708	1.3	389	0.5	385	0.4
RC103	531	1.5	308	0.8	314	0.5
RC104	422	1.5	305	0.5	320	0.4
RC105	845	1.3	531	0.7	402	0.5
RC106	714	1.4	400	0.9	400	0.5
RC107	559	1.2	366	0.5	374	0.5
RC108	425	1.2	309	0.5	308	0.5

Table 25: Results for ILSSR with H = 50 for the instances with $\mathcal{N}$ = 25

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	438	1.0	238	0.5	237	0.4
C102	404	1.0	215	0.6	215	0.5
C103	374	0.7	217	0.4	246	0.4
C104	353	0.6	196	0.4	195	0.4
C105	438	1.0	239	**0.6**	239	0.5
C106	418	1.3	230	0.5	231	0.5
C107	429	1.0	235	0.6	235	0.5
C108	412	1.2	236	0.5	228	0.4
R101	1023	2.3	881	1.4	741	1.0
R102	874	2.7	724	1.6	578	1.0
R103	768	6.8	592	3.6	495	2.4
R104	715	6.7	583	3.2	403	1.6
R105	891	6.2	724	2.9	568	1.4
R106	823	7.8	618	2.9	511	1.3
R107	680	2.9	489	1.3	389	0.6
R108	667	3.5	489	1.5	392	6.1
RC101	842	1.0	559	0.6	524	0.4
RC102	845	1.2	387	0.5	388	0.4
RC103	418	1.6	311	0.5	313	0.5
RC104	422	2.4	305	0.5	313	0.4
RC105	844	1.2	553	0.6	454	0.5
RC106	720	1.4	536	0.8	399	0.5
RC107	556	1.3	366	0.6	372	0.5
RC108	446	1.2	314	0.4	324	0.4

Table 26: Results for ILSSR with H = 100 for the instances with $\mathcal{N}$ = 25

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	432	3.9	316	1.9	238	1.7
C102	402	4.0	226	2.4	215	1.6
C103	375	0.6	220	0.5	233	0.5
C104	357	0.6	198	0.4	202	0.5
C105	428	1.2	315	0.6	235	0.5
C106	418	1.7	307	0.7	235	0.5
C107	427	1.4	318	1.5	235	0.6
C108	415	1.6	307	1.0	245	0.5
R101	1019	2.5	890	1.7	749	1.0
R102	876	2.8	732	1.7	577	1.1
R103	808	3.0	621	1.7	504	0.7
R104	708	2.9	583	1.1	410	0.6
R105	903	2.7	715	1.5	575	0.8
R106	783	2.6	624	1.5	515	0.6
R107	737	3.0	487	1.3	385	0.5
R108	665	3.5	489	1.4	390	0.5
RC101	845	1.2	684	0.7	523	0.5
RC102	716	1.2	389	0.6	389	0.6
RC103	530	1.6	316	0.6	322	0.5
RC104	547	1.7	308	0.5	309	0.5
RC105	848	1.3	599	0.7	460	0.4
RC106	724	1.4	551	0.8	405	0.4
RC107	561	1.7	374	0.8	381	0.7
RC108	544	1.5	316	0.5	365	0.4

Table 27: Results for ILSSR with H = 250 for the instances with $\mathcal{N}$ = 25

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	453	1.5	362	0.6	317	0.5
C102	400	1.2	293	0.5	215	0.5
C103	410	0.9	299	0.5	217	0.4
C104	362	0.8	281	0.6	206	0.4
C105	508	1.4	334	**0.6**	235	0.5
C106	411	1.7	307	1.2	229	1.1
C107	426	1.3	323	0.6	238	0.5
C108	408	1.1	319	0.6	248	0.5
R101	1034	8.6	923	6.2	808	0.4
R102	894	3.0	754	2.1	643	1.4
R103	837	3.2	642	1.8	522	0.8
R104	715	2.9	603	1.7	522	0.8
R105	892	2.7	742	1.7	602	0.9
R106	836	2.8	699	1.6	525	0.8
R107	744	2.9	571	1.6	472	0.8
R108	676	4.5	571	1.9	468	0.8
RC101	860	1.4	638	0.7	532	0.4
RC102	836	1.6	473	0.6	389	0.5
RC103	548	1.8	313	0.7	322	0.5
RC104	518	2.3	320	0.7	324	0.4
RC105	833	1.4	575	0.8	460	0.5
RC106	715	1.6	517	0.9	406	0.6
RC107	600	1.8	473	0.8	381	0.5
RC108	527	1.8	317	1.0	332	0.5

Table 28: Results for ILSSR with H = 10 for the instances with $\mathcal{N} = 50$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	626	6.9	464	2.9	464	2.6
C102	618	7.1	449	2.8	460	2.7
C103	618	7.6	470	3.0	495	2.6
C104	567	5.8	410	2.5	432	2.2
C105	622	6.2	451	2.8	459	2.6
C106	643	6.4	448	2.8	458	2.6
C107	621	6.7	462	2.9	477	2.7
C108	629	8.2	498	3.0	482	2.9
R101	1710	11.7	1443	7.5	1188	3.9
R102	1562	10.6	1206	7.8	1019	3.1
R103	1249	12.3	965	9.2	799	2.7
R104	1209	10.7	841	5.4	724	2.7
R105	1536	12.1	1180	6.5	884	3.0
R106	1383	12.2	1031	6.3	795	2.6
R107	1136	13.3	793	6.3	697	3.1
R108	1067	15.1	717	4.9	677	2.9
RC101	1717	6.5	1049	2.7	970	2.2
RC102	1398	6.8	886	2.5	944	2.2
RC103	1273	6.2	747	2.5	764	2.2
RC104	909	6.5	730	2.6	704	2.4
RC105	1320	6.6	876	2.5	893	2.3
RC106	1418	6.3	931	2.5	939	2.3
RC107	1121	5.7	771	2.8	806	2.6
RC108	1009	6.1	668	2.5	731	2.4

Table 29: Results for ILSSR with H = 50 for the instances with $\mathcal{N} = 50$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	621	6.2	459	3.0	457	2.6
C102	616	7.0	461	2.8	438	2.4
C103	638	8.6	445	2.8	511	2.6
C104	590	6.4	414	2.5	427	2.5
C105	622	6.7	452	2.8	444	2.6
C106	711	6.7	472	2.9	462	2.6
C107	621	7.3	464	3.2	452	2.6
C108	623	8.1	449	3.6	514	2.6
R101	1753	11.1	1435	6.7	1189	3.4
R102	1566	10.6	1261	7.5	979	3.3
R103	1249	25.2	947	12.5	798	6.0
R104	1112	20.4	822	11.2	728	6.7
R105	1549	2.4	1206	19.7	888	7.6
R106	1349	21.7	1063	7.1	814	2.5
R107	1148	13.5	828	5.7	706	2.7
R108	1066	14.0	715	5.7	662	3.2
RC101	1682	6.5	1064	2.9	974	2.3
RC102	1415	6.5	1013	2.9	902	2.4
RC103	1208	6.6	740	2.4	739	2.3
RC104	934	6.4	701	2.5	742	2.4
RC105	1316	6.9	827	2.5	896	2.3
RC106	1421	6.2	1015	2.6	942	2.3
RC107	1254	5.5	787	2.4	791	2.5
RC108	1027	4.9	701	2.3	773	2.3

Table 30: Results for ILSSR with H = 100 for the instances with $\mathcal{N} = 50$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	701	19.3	554	8.7	453	8.8
C102	708	15.4	458	4.3	460	3.2
C103	701	9.2	451	4.7	458	3.2
C104	574	7.7	408	3.3	433	3.3
C105	702	8.3	444	3.6	457	2.8
C106	704	9.0	471	3.6	457	3.1
C107	702	9.2	461	3.9	477	2.5
C108	707	10.2	470	14.0	465	8.4
R101	1719	11.1	1417	7.3	1186	3.9
R102	1592	10.7	1242	8.7	1031	3.7
R103	1263	12.6	954	7.0	770	3.1
R104	1186	10.3	893	4.9	686	2.6
R105	1526	11.5	1212	6.4	885	3.2
R106	1317	10.9	1054	6.4	825	2.5
R107	1148	12.4	799	6.0	693	2.6
R108	1048	14.1	723	5.7	681	2.8
RC101	1775	7.2	1058	3.1	983	2.4
RC102	1458	8.8	1034	3.1	954	2.3
RC103	1290	7.1	769	2.5	775	2.2
RC104	918	6.5	641	2.6	695	2.4
RC105	1381	6.8	866	2.7	849	2.2
RC106	1428	6.6	1025	2.6	936	2.3
RC107	1105	6.8	817	2.5	810	2.3
RC108	1054	5.9	710	2.4	774	2.4

Table 31: Results for ILSSR with H = 250 for the instances with $\mathcal{N} = 50$

	$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$	
Instance	Costs	Time(ms)	Costs	Time(ms)	Costs	Time(ms)
C101	800	9.9	604	5.5	462	3.1
C102	738	10.2	528	5.0	448	2.8
C103	684	9.7	537	4.1	471	2.7
C104	686	8.2	502	3.8	430	2.6
C105	773	9.5	604	5.0	450	2.7
C106	805	11.7	535	4.7	476	2.6
C107	719	9.9	524	5.1	458	2.8
C108	788	8.7	630	4.9	464	2.8
R101	1802	37.2	1508	8.7	1275	5.3
R102	1660	12.8	1362	9.2	1094	4.2
R103	1308	16.5	1032	8.8	880	4.0
R104	1245	14.9	945	8.3	701	3.4
R105	1563	12.7	1256	7.5	1000	4.0
R106	1425	13.2	1139	7.7	902	3.4
R107	1217	14.9	864	8.9	694	3.2
R108	1150	18.8	882	8.7	676	3.0
RC101	1578	7.2	1155	3.7	990	2.2
RC102	1447	7.4	1107	3.0	937	2.3
RC103	1278	8.0	714	3.0	779	2.3
RC104	1017	7.4	717	2.8	713	2.5
RC105	1361	6.9	972	3.1	841	2.5
RC106	1497	8.6	1051	4.2	900	2.5
RC107	1223	6.4	916	2.7	806	2.4
RC108	1193	6.8	798	2.6	750	2.3

D Programming Code

For obtaining results for the inter local search metaheuristic and the inter local search with steep routes metaheuristic, we use Java code. The code is described below.

- replication/LocalSearchHeuristic.java This is a code for obtaining the routes and costs for every instance for the ILS. As input the cluster is set equal to "c", "r" or "rc". Furthermore, the code will run for 8 instances and for every cluster size (N = 10, 15, 25, 50 and 100). This is the main code for the ILS and it makes use of the Route class, ConventionalRoute class, ElectricalRoute class and the Solution class. Before provide the
- replication/Route.java All information of one route is stored in the Route class in the initialization phase of the ILS. The Route class stores all information of the customers that are served in that route and has methods to validate feasibility, determine the best recharge station etc.. It stores both routes for conventional vehicles as for electrical vehicles
- replication/ConventionalRoute.java For the local search phase, the routes executed by conventional vehicles are stored in a ConventionalRoute object. It stores all information of the customers and includes methods to validate the feasibility.
- replication/ElectricalRoute.java For the local search phase, the routes executed by electrical vehicles are stored in a ElectricalRoute object. It stores all information of the customers and includes methods to validate the feasibility.
- replication/Solution.java All ConventionalRoute objects and ElectricalRoute objects are stored in a Solution object. Furthermore, it stores the costs of the solution and the total emission emitted by all ConventionalRoute objects. Besides containing information about the solution of an iteration, the Solution object includes a method for the local search and the perturbation.
- extension/SummaryStatistics.java For obtaining information of the CO_2 emission per km and energy consumption over all instances we apply this code for each cluster "c", "r" or "rc".
- extension/generateHeights.java This code generates the heights where each depot, recharge station and customer is located for each instance. It stores a value between 0 and 1 in a txt-file.
- extension/LocalSearchHeuristic.java This is the main code for obtaining the routes and costs for every instance for the ILSSR. The heights of every location are set by multiplying

the value of the generate height in the txt-file with the defined height range. Furthermore, the cluster can be set equal to "c", "r" or "rc". This is the main code for the ILSSR and it makes use of the Route class, ConventionalRoute class, ElectricalRoute class and the Solution class.

- extension/Route.java Same as described in replication.
- extension/ConventionalRoute.java Same as described in replication.
- extension/ElectricalRoute.java Same as described in replication.
- extension/Solution.java Same as described in replication.
- extension/NewCostWithSR For computing the new costs for the obtained routes where a flat area was assumed we use this code to compute the costs when height differences occur and CO_2 emission and energy consumption differ. As input can the cluster type be set equal to "c", "r" or "rc".