Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics & Operations Research

Modelling different functions for partial recharging and pollution in the Mixed-Fleet Green Vehicle Routing Problem with Partial Recharging and Time Windows

Iris Nieuwland (581577)

Supervisor:	Ymro Hoogendoorn
Second assessor:	Riley Badenbroek
Date final version:	2nd July 2023

Abstract

In this research, we investigate the effect of different functions for charging time and pollution rate in the Mixed Fleet Green Vehicle Routing Problem with Partial Charging and Time Windows with an upper bound on pollution emission. We use the Iterated Local Search metaheuristic to examine the effect of these different functions in terms of feasibility. The mixed fleet consists of electrical and conventional vehicles and recharging on the route is allowed for electrical vehicles. For the charging time, we examine a function that is linear and one that is non-linear in the amount recharged. When linear is assumed but non-linear holds, we see that 27.5% of the feasible solutions become infeasible. Furthermore, for small instances, assuming the non-linear function leads to a decrease in distance of 11.4% in comparison to assuming the linear function. For the pollution rate, we compare a function that is piecewise constant in the load of the vehicle and a linear one. We find that the total pollution is lower for the linear function than for the piecewise constant function, which means that assuming the piecewise constant function does not cause infeasibility for our test instances.

1 Introduction

Almost 21% of all pollution is caused by road transport (Zhang & Batterman, 2010). In order to reduce these pollution emissions, transport strategies must be chosen more sustainably instead of just as cheaply as possible, which changes routing problems.

This research aims to investigate how the travelled distance can be minimized while reducing pollution. In particular, we aim to keep the total pollution below a certain level, as in Macrina et al. (2019). The problem is called a Green Vehicle Routing Problem (G-VRP), where we minimize the total distance travelled to serve customers from a depot with either conventional (ICCVs) or electrical (ECVs) vehicles, of which the last one can be recharged throughout the route. Furthermore, we want to keep the pollution emissions below a certain level.

ICCVs produce pollution, so ECVs are needed to make the problem feasible when the upper bound constraint is binding. Because ECVs have a low range, these vehicles can charge on the route. Therefore, they take longer to travel and they drive more kilometres if the charging station is not right on the route. In contrast, ECVs do not emit any pollution. We thus want to find a good balance between using ICCVs and ECVs. We do allow for partial recharging. Furthermore, time windows are taken into account, such that all customers are helped within their desired time interval.

We investigate the effect of using different functions for the charging time. Therefore, we investigate two functions, one that is linear and one that is non-linear in the amount of recharged energy. We study if the solution remains feasible when the linear function is assumed, but the non-linear function holds. When this is the case we also look at the effect on the distance when this solution is compared to the solution when non-linear charging is assumed.

Furthermore, we study the effect of using different functions on the pollution rate. Here, we focus on a function that is piecewise constant in the load percentage and a function that is linear in this percentage. We again investigate the effect of assuming the piecewise constant function when the linear function holds on the feasibility of the solution.

To generate a solution to this problem, we use the iterated local search (ILS) metaheuristic of Macrina et al. (2019), because they state that solving the problem to optimality is not possible between reasonable computation time. In this metaheuristic, we generate an initial solution, and then we iteratively apply a certain perturbation and the local search procedure, where customers are moved from one route to another in the best way possible.

This paper proceeds as follows. In Section 3, we describe the problem of this research clearly. In Section 2, an overview of previous research on this subject is given. The methodology and associated mathematical formulations are described in Section 5. In Section 6, the test instances for the problem and the results are given and lastly, a conclusion is drawn in Section 7.

2 Literature review

A lot of research has already been done on how Vehicle Routing Problems (VRPs) change when environmental pollution is included in the model.

The problem in this study is based on the Vehicle Routing Problem (VRP) as described in the literature. It is stated by Toth & Vigo (2002) that a VRP is a problem in which the goal is to find an optimal set of routes to be driven to serve particular customers. Optimality can be different among different goals of research, for instance, the costs can be minimized, the distance driven can be minimized or the satisfaction of customers can be maximized. Because environmental friendliness has become an increasingly important aspect in society, Bektaş & Laporte (2011) investigated how, in addition to minimizing the distance travelled, gas emissions can be included in the VRP. They created a model in which minimal pollution is balanced against minimal costs. They call this problem, in which only conventional vehicles are used, the Pollution Routing Problem (PRP). Through a collaboration with Emrah Demir, it became clear how difficult it is to solve a problem like the PRP to optimality. Demir et al. (2012) proposed an Effective Adaptive Large Neighborhood Search (ALNS) heuristic, with which they solve the problem following an iterative process.

Erdoğan & Miller-Hooks (2012) formulate a Green-VRP (G-VRP), in which only electric vehicles are used in the fleet. They minimize the total distance travelled through two heuristics and prove the usefulness of these heuristics. In the first heuristic, the Modified Clarke and Wright Savings (MCWS) Heuristic, a solution is generated from routes that go to a single customer and back. The second heuristic called the Density-Based Clustering Algorithm (DBCA), is based on the number of customers within certain areas. These heuristics are especially useful when few charging stations are available, charging times are assumed to be linear and only full recharging is possible.

Koç et al. (2014) investigated and proved the utility of introducing a heterogeneous fleet consisting of conventional vehicles with different sizes, by solving the Fleet Size and Mix Pollution-Routing problem, which is an extension of the PRP by considering this heterogeneous vehicle fleet. It is clearly shown in this study that the use of different types of vehicles can be beneficial compared to using a single type.

Felipe et al. (2014) investigate the effect on the G-VRP when partial charging is introduced and when different charging technologies, so faster, more expensive or slower, cheaper charging, are examined. They conclude that partial recharging provides significantly lower costs and energy savings. Furthermore, none of the different charging technologies performed better for all instances, which emphasizes the utility to use several charging technologies.

Sassi et al. (2014) investigate the use of both conventional and electric vehicles on the G-VRP. In addition, they use time and location-dependent charging costs. They first try to use as few vehicles as possible, after which they want to minimize the kilometres driven by these vehicles. They develop a Charging Routing Heuristic, to create an initial solution that uses as few vehicles as possible. In addition, they design an Inject-Eject-Based Local Search heuristic to best solve problems of this type, in which customers are iteratively moved in a random way. To be able to use a charging station several times, dummy nodes are often required. Koç & Karaoglan (2016) introduce new decision variables, eliminating the need for dummy nodes in the G-VRP.

Also Montoya et al. (2016) did additional research following the G-VRP. By creating a multi-space sampling heuristic, they show that this heuristic is the best and easiest method to solve the G-VRP with constant refuelling time and a homogeneous fleet of only electric vehicles. Montoya et al. (2017) state that vehicle charging is non-linear, and that assuming linearity can lead to expensive and even infeasible outcomes. Macrina et al. (2019) examined the G-VRP with a fleet of both conventional and electrical vehicles in which they introduce a limit on the total pollution. They provide a metaheuristic in which they assume a linear charging function and a piecewise constant pollution rate per km. Lastly, Kancharla & Ramadurai (2020) provide a model in which both recharging and the consumption of energy is approached non-linearly. They provide a mixed-integer linear program (MILP) formulation and an ALNS heuristic to determine the best route and charging strategy.

3 Problem description

In this section, we describe the Mixed Fleet Green Vehicle Routing Problem with Partial Recharging and Time Windows of Macrina et al. (2019). The problem consists of a depot s, a set of customers $\mathcal{N}$ and a set of charging stations $\mathcal{R}$. All elements in both sets and the depot, have certain coordinates (x_i, y_i). Because we have a mixed fleet, consisting of conventional and electrical vehicles, ICCVs and ECVs respectively, there are two different kinds of routes: routes that are served by an ICCV and routes that are served by an ECV, which can be recharged on the route. We want to minimize the total distance d travelled.

A feasible solution consists of two sets of routes: the set C, the set of routes driven by an ICCV, and the set E, the set of routes driven by an ECV. All routes have to start and end at the depot and all customers have to be served exactly once. All customers have a demand q_i that must be met and there is a constraint on the capacity Q of the vehicle. The total demand of the customers in a route can not be higher than the total capacity of the vehicle.

Each customer has a certain time window $[e_i, l_i]$. Here, e_i indicates when that particular customer is ready to be served. From this time and later, the customer can be visited. If the vehicle is too early, it must wait until e_i to serve the customer. l_i indicates the end time, which is the last moment for a service to start. Furthermore, all customers have a service time s_i, the time it takes to serve the customer.

In addition to customers having a time window, the problem also has a time window $[e_s, l_s]$. Vehicles may only be on the road within this time. Charging stations do have a service time dependent on the amount recharged. The recharging time can be calculated via a formula τ. When charging ECVs, it must be taken into account that the energy in the vehicle is enough to complete the route or reach the next charging station. In addition, the vehicle has a certain maximum energy capacity B, which may not be exceeded. Charging is required at the depot to start the route.

Finally, we set an upper bound UB on the total pollution Φ that can be caused by the problem. Φ is the sum of all pollution caused on arcs in the route, which can be calculated separately with a formula ϕ,

4 Different functions for charging time and pollution rate

To indicate the effect of different charging and pollution functions, we need to define these functions. In Section 4.1 two functions for the charging time τ are defined: a linear and a non-linear function. Furthermore, Section 4.2 defines a piecewise constant and a linear function ϕ for the pollution rate.

4.1 Charging time: Linear vs non-linear

As in Macrina et al. (2019), the first charging function τ is linear. Each problem has its inverse charging rate that indicates how much time it takes to charge one unit of energy.

Montoya et al. (2017) argue that the assumption of linear charging times can lead to expensive or even infeasible solutions to the problem. The speed of charging may be different when the battery is almost empty than if it is almost full. They state that there are linear segments in the charging function, but the slope differs for each segment in a convex way. In particular, a better, non-linear charging function can often be allocated into three different segments: the first has the flattest slope and the last has the steepest slope. This holds when the battery level (on the x-axis) is plotted against the time (on the y-axis). It means that charging happens faster in the beginning than in the end. This is the second proposed charging function τ.

To test the effect of the different functions in the most extreme way, we assume that the linear charging function used in Macrina et al. (2019) is the most optimistic charging time possible, because, in this way, the effect in the worst case is examined. As in Montoya et al. (2017), the non-linear function consists of three linear functions, separated by segments, where the function in the first segment is similar to the completely linear function. In the second segment, charging takes twice as much time per unit, so the slope is half of what it was in the first segment. Lastly, in the third segment, the time to charge one unit of energy is three times as large as in the first, so the third slope is one-third of the first slope.

The segments are dependent on the total energy capacity of the vehicle. Again, following Montoya et al. (2017), charging the vehicle up to 70% of its full capacity follows the linear function in the first segment, so charging small values is similar to what is used in the linear approach. After that, between 70 and 85%, the recharging follows the function in the second segment. Furthermore, between 85 and 100%, the function used in the one in the third segment.

Both functions are visible in Figure 1. The inverse charging rate is assumed to be 3.47 for this figure, but it may differ for different test instances.

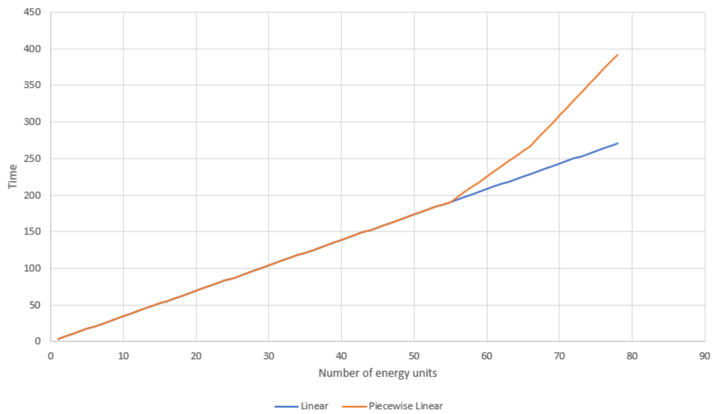

Figure 1: The linear and piecewise linear charging functions τ

For the charging times, we compare the results of the linear charging function to the non-linear charging function. To perform this comparison, we generate the solution using the linear charging function. After that, we verify if the solution is still feasible when using the non-linear function. The infeasibility of the solution can only be caused by exceeding the time window constraints, as the different charging function is only affecting that part.

For the solutions that remain feasible, we generate a new solution using the non-linear charging function and investigate the effect this had on the obtained distance of the best solution.

4.2 Pollution rate: Piecewise constant vs linear

The first function ϕ for pollution is piecewise constant as in Macrina et al. (2019). The pollution per km is dependent on the load percentage of the vehicle, which is defined as the load divided by the total capacity of the vehicle, the values for the pollution rate per km are given in Table 1.

Weight laden (%)	Pollution rate (kg CO_2/km)
0 - 25	0.77
26 - 50	0.83
51 - 75	0.90
76 - 99	0.95
100	1.01

Table 1: Pollution rates for different vehicle loads

In this piecewise constant structure, it happens to be that the pollution of a vehicle with a load that takes 25% of the total load capacity has the same pollution rate per km as a vehicle that is empty, while this rate does differ a lot when comparing a 25% load to a 26% load. In the linear function we propose, the difference between 0 and 25% is seen as larger than between 25 and 26%. The second function of ϕ follows from formulating the function as the linear trendline of the values of the piecewise constant function used before, this function is equal to (1). Both functions, piecewise constant and linear, are visible in Figure 2.

$$y = 0.0023x + 0.7458 \tag{1}$$

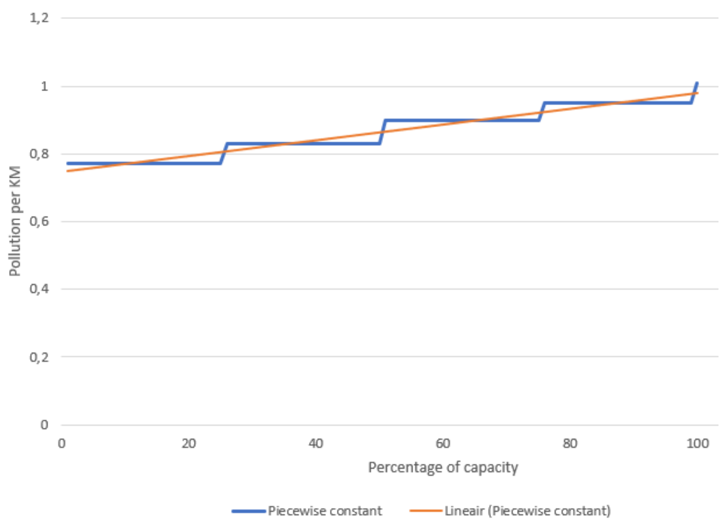

Figure 2: The piecewise constant and linear pollution functions ϕ

To indicate the effect of this function on total pollution, we generate a solution using the piecewise constant function. Based on this solution, we recalculate the total pollution and we perform another feasibility test, this time we check if the total pollution is still lower than our desired upper bound.

5 Methodology

We solve our problem with the Iterated Local Search metaheuristic as proposed by Macrina et al. (2019), where an upper bound on the best solution is generated. This is done via multiple components, which are all described in this section. In Section 5.1, we describe the general metaheuristic. Furthermore, in Section **??** and **??**, the two components, the initialization and the local search procedure, of the metaheuristic are expressed as separate heuristics.

5.1 Iterated local search metaheuristic

As Macrina et al. (2019) states, solving the full mathematical problem to optimization is not possible within decent computation time. Therefore, we use their proposed metaheuristic to solve the problem. Via this heuristic, the problem is not necessarily solved to optimality, but it will find a feasible solution that is acceptable. The metaheuristic is based on iterated local search (ILS), where we first generate an initial solution, whereafter we apply a particular local search (LS) procedure to it. We address a certain perturbation and this local search procedure until a stop criterion is met. This general algorithm for the ILS is shown in Algorithm 1.

Algorithm 1 `iteratedLocalSearch`

Output: A best solution to the problem: η^*

```
η0 ← generateInitial                          ▷ Algorithm 2
η1 ← localSearch                              ▷ Algorithm 6
while Stop criterion not satisfied do
    perturbation
    localSearch → ηk
    if ηk is better than η* then
        η* ← ηk
    end if
end while
```

In this case, the stop criterion is set to a fixed number of iterations, namely 1,000,000. Furthermore, a worsening of the solution is used as the perturbation. The perturbation involves choosing a random customer and a random route, whereafter the customer is removed from its current route and added to the randomly chosen route on another random position. The resulting solution does not need to be feasible, as the local search heuristic takes care of that.

In the end, as the solution to our problem, we choose the solution in which the lowest distance is obtained η^*. Furthermore, when we cannot find any feasible solution, we choose the solution in which the pollution violation is as low as possible.

5.1.1 Initialization

To generate an initial solution, we use the sequential insertion heuristic (SIH) as in Macrina et al. (2019). The general algorithm of the SIH is given in Algorithm 2.

Algorithm 2 `generateInitial`

Output: An initial solution for the ILS: η_0

  `cluster` $\longrightarrow E', C'$ ▷ Algorithm 3
  `insertionConventional` $\longrightarrow \eta_C$ ▷ Algorithm 4
  if There are customers left in C' **then**
    Update $E' = E' \cup \mathcal{N}^-$, where $\mathcal{N}^-$ are all unrouted customers
  end if
  `insertionElectrical` $\longrightarrow \eta_E$ ▷ Algorithm 5
  if There are customers left in E' **then**
    Update $C' = C' \cup \mathcal{N}^-$, where $\mathcal{N}^-$ are all unrouted customers and relax the upper bound on pollution
    `insertionConventional` without the upper bound $\longrightarrow \eta_C$
  end if
  $\eta_0 = \eta_C \cup \eta_E$

Here, all customers in $\mathcal{N}$ are divided into two groups, a group C' that is served by ICCVs, and a group E' that is served by ECVs. This division is made on the basis of a clustering algorithm, described in the paragraphs later in this section. When these groups, or clusters, are known, we use the insertion heuristics to insert all customers that are not allocated to a route yet, to a route. We start with the conventional cluster C' and use the insertion heuristic for conventional routes to insert all customers into a route. Sometimes insertion will cause infeasibility while creating a whole new route is not profitable with eyes on the option of using an ECV instead. All customers for whom this is the case are added to the electrical cluster E'. After this, we use the insertion heuristic for all customers in E' to allocate all other customers to a route. We end with the wanted initial solution. If it is not possible to add all these customers to an electrical route, we allow the pollution upper bound to be exceeded and add the customer to a conventional route anyway. For routes driven by an ICCV, a different heuristic is used than for the routes driven by ECVs. These heuristics are described later in this section.

Clustering algorithm As said before, we need two clusters of customers to create an initial solution for our heuristic, one consisting of customers that will be served by ICCVs (C') and one consisting of customers served by ECVs (E'). The clusters have to be formed in a way such that all customers are divided along sets C' and E'. To create these sets, we start with two sets E and C both initialized as the set of only the starting node s, so $E = s$ en $C = s$. To choose which customer is added to which set, we calculate some scores $1 \leq p_i^E \leq 10$ and $1 \leq p_i^C \leq 10$. For the set of vertices visited by ECVs, we calculate p_i^E as in Macrina et al. (2019), as given in (2). Here, d_i^E denotes the distance from the barycentre of E (b_E) to customer i. This barycentre is based on the locations of all customers that are already in E at that certain time. Furthermore, d_{min}^E and d_{max}^E denote the Euclidean distance from b_E to the nearest and furthest customer out of all customers, respectively.

$$p_i^E = 11 - \left(1 + \frac{d_i^E - d_{min}^E}{d_{max}^E - d_{min}^E} \times 9\right) \tag{2}$$

For the set of customers that are served by an ICCV, we use the formula in (3). In this formula, λ is a constant between 0 and 1, we use $\lambda = 0.5$. Furthermore, $\text{pDist}_i^C = 11 - \left(1 + \frac{d_i^C - d_{min}^C}{d_{max}^C - d_{min}^C} \times 9\right)$, and $pQ_i^C = 11 - \left(1 + \frac{q_i - q_{min}}{q_{max} - q_{min}} \times 9\right)$. The distance d_i^C, d_{min}^C and d_{max}^C are now the distances from the barycentre of C (b_C), computed in the same way as b_E. Furthermore, q_i, q_{min} and q_{max} denote the demand of customer i, the smallest customer demand and the largest customer demand of all customers in $\mathcal{N}$ respectively, even when a customer is already allocated to a route.

$$p_i^C = \lambda(\text{pDist}_i^C) + (1 - \lambda)(\text{pQ}_i^C) \tag{3}$$

When these scores are known, we can choose which customers we add to which cluster. For every iteration, we look at the customer with the largest score to each cluster, i.e. $i_E^* = \text{argmax}_{i \in \mathcal{N} \setminus C' \cup E'}\{p_i^E\}$ and $i_C^* = \text{argmax}_{i \in \mathcal{N} \setminus C' \cup E'}\{p_i^C\}$. When the customer with the highest score is different, i.e. $i_E^* \neq i_C^*$, we can add both customers to the cluster of which they obtained the highest score. However, if the highest score for both of the clusters is obtained for the same customer, we assign the customer to E if $p_{i_E^*}^E > p_{i_C^*}^C$ and to C otherwise. When this allocation is done, we can recalculate the barycentre and all scores. We do this until all customers are in a cluster. When this is the case, we remove the starting node s and we have our desired clusters E and C. A pseudocode for this algorithm is given in Algorithm 3.

Algorithm 3 `cluster`

Input: All customers in $\mathcal{N}$
Output: all customers divided into two clusters: E' and C'
  Initialize $E = \{s\}$ and $C = \{s\}$
  while There are customers left **do**
    Calculate barycentres b_E and b_C
    Calculate p_i^E and p_i^C for all customers that are not allocated ▷ (2) and (3)
    Define $i_E^* = \text{argmax}_{i \in \mathcal{N} \setminus C' \cup E'} \{p_i^E\}$ and $i_C^* = \text{argmax}_{i \in \mathcal{N} \setminus C' \cup E'} \{p_i^C\}$
    if $i_E^* \neq i_C^*$ **then**
      Add i_E^* to E and i_C^* to C
    else if $p_{i_E^*}^E > p_{i_C^*}^C$ **then**
      Add i_E^* to E
    else
      Add i_E^* to C
    end if
  end while
  Remove s from both sets E and C

Insertion heuristic for conventional routes The strategy to insert customers in routes driven by ICCVs starts with initializing a route Z_0^C from the starting node of the depot, then to the customer $i' \in C'$ with the lowest end time $l_{i'}$ in the time window and closing with the end node of the depot. After initializing, we can add other customers to the route. Let us take the current route $Z_k^C = (s, i_1, i_2, ..., i_m, s)$ and assume that we want to add a customer to this route. To choose which customer is added, we calculate the best position $f_1(i(u), u, j(u))$ in terms of distance for every unrouted customer $u \in C'$. The best position is the position which causes the least extra distance to add the customer, see (4). Here, $i(u)$ and $j(u)$ are two adjacent vertices in the route, and $f_((i_{p-1}, u, i_p)$ is the added distance when u is inserted between node i_{p-1} and i_p.

$$f_1(i(u), u, j(u)) = \min_{p=1,...,m}\{f_1(i_{p-1}, u, i_p)\} \tag{4}$$

For every unrouted node, we then compare the costs of adding the customer to a new route to the costs of adding the customer to the existing route. We do this by subtracting these two values as in (5). Here $c_{s,u}^C$ is the cost, in terms of distance, of travelling from s to u.

$$f_2(i(u), u, j(u)) = c_{s,u}^C - f_1(i(u), u, j(u)) \tag{5}$$

The unrouted customer for which the value of f_2 is the greatest is the customer that is potentially added to the route, see (6). We take the maximum here because then the cost of adding the customer is the lowest.

$$f_2(i(u^*), u^*, j(u^*)) = \max_u\{f_2(i(u), u, j(u))\} \tag{6}$$

Before adding the customer to the route, we have to verify that adding is feasible in terms of pollution and time windows. When it is feasible, the customer is added to the route. Else, when adding the customer to a new route, create a new route and add the customer to it. When both are not the case, we add the customer to E' and let it be served by an ECV. The full algorithm is summarized in Algorithm 4.

Algorithm 4 `insertionConventional`

Input: All customers in C'
Output: Routes for ICCVs
  Initialize $Z_0^C = (s, i', s)$ with $i' \in C'$ the unserved customer with the smallest end time $l_{i'}$
  while There are customers in C' left **do**
    Calculate $f_1(i(u), u, j(u))$ for every unserved customer $u \in C'$
    Calculate $f_2(i(u^*), u^*, j(u^*))$
    if Adding u^* between $i(u^*)$ and $j(u^*)$ is feasible **then**
      Add u^* on that position in the route
    else if Adding u^* to a new route is feasible **then**
      Add u^* to a new route Z_{k+1}
    else
      Add u^* to E'
    end if
  end while

Insertion heuristic for electrical routes The strategy of inserting a customer into an electrical route starts the same as for conventional routes. We initialize a route $Z_0^E = (s, i', s)$, with i' the unserved customer in E' with the smallest end time. After that we calculate $f_2(i(u^*), u^*, j(u^*))$ using formulas (4), (5) and (6), but now with the costs of ECVs, $c_{i,j}^E$, which are again similar to distance.

Inserting customers into a route is a bit more difficult than in conventional routes, as energy is an extra factor in the problem. Namely, when the energy capacity constraint is exceeded in a route, a charging station has to be added. We do this by checking if all the customers can be reached with the remaining energy in the vehicle. For this, we start at the depot and go through all the customers in the route one by one. When a customer cannot be reached, we check if the closest charging station from the previous customer can be reached. When this is the case, we add this charging station to the route and recharge the vehicle to its full capacity. When this charging station cannot be reached, we try to reach the closest station from the customer before and so on. When the route is feasible in energy again, we lower the amount of charged energy to the minimum that is needed, so the amount of energy needed to reach the next charging station in the route or the depot. This algorithm to add charging stations is the same every time it is needed to add one.

When the route is completely feasible, we add it to the solution. When the route is only feasible in capacity and time, but not in energy, we add charging stations to the route. When after this the time constraint is not satisfied anymore, we remove random customers and reallocate the charging stations on the route until the route is completely feasible. We choose the customer that has to be removed randomly out of all customers in the route and reallocate the charging stations on the route. The removed customers in this state are added to the conventional cluster C, which indicates that the pollution emission constraint is relaxed in the initialisation. However, when the route is feasible in energy, we check if there are still customers left in E and initialize a new route in the same way we did before.

When the heuristic cannot find a feasible solution, we add all unserved customers to C' again and allow the emission constraint to be violated. The initial solution is not feasible in that case, but that infeasibility is fixed in the LS procedure. A summary of the insertion heuristic for electrical route can be found in Algorithm 5.

Algorithm 5 `insertionElectrical`

Input: All customers in E'
Output: Routes for ECVs
Initialize $Z_0^E = (s, i', s)$ with $i' \in E'$ the unserved customer with the smallest end time $l_{i'}$
while There are customers in E' left **do**
  Calculate $f_1(i(u), u, j(u))$ for every unserved customer $u \in E'$
  Calculate $f_2(i(u^*), u^*, j(u^*))$
  if Adding u^* between $i(u^*)$ and $j(u^*)$ in the current route is feasible in capacity and time **then**
    if Route is infeasible in energy **then**
      Add the nearest charging stations to the route and recharge the vehicle
      if Time window constraint is satisfied but some customers are unserved **then**
        Add customer u^* to the current route between $i(u^*)$ and $j(u^*)$
      end if
    else if There are unserved customers left **then**
      Add u^* to the route
      Initialise a new route with the customer with the smallest end time left in the cluster
      Add current route to solution and take the new route as the current
    end if
  else if Adding u^* to a new route is feasible **then**
    Add u^* to a new route
    Add current route to solution and take the new route as the current
  else
    Add u^* to C' and violate pollution constraint
  end if
end while

5.1.2 Local search procedure

The main procedure of the ILS metaheuristic is the local search procedure (LS). This procedure is intended to improve an initial solution to make it more optimal. The procedure is based on two different improvement heuristics. When the initial solution is already feasible, we apply a basic improvement heuristic and when the initial solution is not feasible we apply an improvement heuristic with a penalty function.

Algorithm 6 `localSearch`

Input: A starting solution η'
Output: A solution η_k
if η' is feasible **then**
  Apply improvement heuristic $\longrightarrow \eta_k$
else
  Apply improvement heuristic with penalty $\longrightarrow \eta_k$
end if

Improvement heuristic In the improvement heuristic, we relocate some customers. There are three possibilities, a change that affects two conventional routes, two electrical routes or one conventional and one electrical route. In every situation, we consider every customer and check if they can be feasibly inserted into another route. We consider every possible position of insertion and examine which of these positions is the best for that certain customer. When this insertion is feasible, we relocate the customer to that new position. It is worth mentioning that if electrical routes are involved, the insertion can cause the need for an extra charging station. The other way around, removing a customer from an electrical route can cause the possibility to remove a charging station from the route. Therefore, after each improvement, we reallocate the charging stations to the route. We perform this heuristic until no better insertions are feasibly possible.

Improvement heuristic with penalty function In the improvement heuristic with penalty function is infeasibility allowed in the initialization. As follows from previous sections, an infeasible initial solution means that the pollution emission constraints are not satisfied. In this heuristic, the objective function is amended to the function in (7). Here, $z(\eta)$ is the cost function, in our case the total Euclidean distance of the solution, θ is the penalty coefficient, which is initially set to 1, and $e(\eta)$ is the violation of polluting emissions, as stated in (8).

$$z'(\eta) = z(\eta) + \theta e(\eta) \tag{7}$$

$$e(\eta) = \max\left\{0, \sum_{(i,j)\in A} \Phi - UB\right\} \tag{8}$$

As said, initially the penalty coefficient θ is set to 1, but it can change in every iteration. When after the improvement the solution is not feasible yet, we increase the coefficient by 10% and do the improvement with a penalty again, until a feasible solution is found. To get there, we use the improvement heuristic as it is described in the previous paragraph.

6 Results

In this section, we first explain which instances we will use for the numerical study, and explain how they are defined in Section 6.1. We then carry out the numerical investigation and report and analyze the values obtained in Section 6.2.

6.1 Test instances

To test the metaheuristic and the different functions for pollution and charging times, we use the EVRPTW benchmark instances as in Macrina et al. (2019). These benchmark instances are from Schneider et al. (2014), who have adjusted the benchmarks of Solomon (1987). These instances are generated in three different ways: using a clustered distribution of the geographical customer locations, a random distribution or a combination of both. These distributions are referred to as c, r and rc, respectively. Furthermore, we test the instances where 5, 10, 15, 25, 30, 50 or 100 customers have to be served. Here the instances with 25, 30 or 50 customers are generated ourselves by removing the first 75, 70 or 50 customers respectively from the instances with 100 customers. All instances specify the customers and charging stations and their locations. Furthermore, they specify the time windows for all locations, the service time and the desired demand.

We determine the maximum pollution upper bound (UB_{max}) separately for each instance. We do this by first running the program without an upper bound, and then equating UB_{max} to the obtained pollution value, rounded up to the nearest integer. Then, we run the program three more times for this instance with three different upper bound values. In concrete terms, this means that the upper bound is set equal to $0.75 \cdot UB_{max}$, $0.50 \cdot UB_{max}$ and $0.25 \cdot UB_{max}$. The instances (type of distribution and their number), their associated value for the number of customers ($|\mathcal{N}|$), and their version of the pollution upper bound factor (α) are referred to as "TypeNumberC$|\mathcal{N}|$_α". For example, r101C5_0.75 denotes the 101st random distributed instance with 5 customers, with the pollution upper bound equal to $0.75 \cdot UB_{max}$.

Furthermore, each instance specifies the energy and vehicle capacity and the inverse refuelling rate. The energy capacity is the amount of energy that fits into the vehicle at most. The vehicle capacity is the maximum load of units of demand, and the inverse refuelling rate is the time it takes to charge one unit of energy in the linear function. These values are all different for different instances. The inverse refuelling rate does not only affect the linear charging function but also the non-linear charging function as the three sectors in the non-linear function depend on the value as well. For every charging station in the problem, the charging function is equal. This means that charging can be done at the same speed at every charging station.

Moreover, a value for the energy consumption rate and the average velocity is given for all instances, but these are equal to 1 in all cases of this study.

6.2 Numerical results

In total, we have 80 instances: 12 with 5 customers, 12 with 10, 9 with 15, 12 with 25, 11 with 30, 12 with 50 and 12 with 100 customers. For the numerical results, we run the Iterated Local Search for each instance three times. Which means that 240 solutions are generated using the ILS. Table 2 shows the total number of feasible results for different types. The first row indicates how many of the initial solutions were feasible, the second row indicates how many solutions were feasible after running the ILS. Furthermore, the third and fourth rows indicate whether the linear and non-linear time, respectively, is feasible in the solution. Finally, the fifth and sixth rows indicate whether the piecewise constant and linear pollution rate, respectively, are feasible in the solution. The full results for each instance separately can be found in Appendix A until G. All these solutions are based on the linear charging function and the piecewise constant function for the pollution rate.

	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 15$	$\|\mathcal{N}\| = 25$	$\|\mathcal{N}\| = 30$	$\|\mathcal{N}\| = 50$	$\|\mathcal{N}\| = 100$	Total
# Feasible initial solutions	10	3	1	1	0	0	0	15
# Feasible ILS solution	30	29	17	16	10	6	8	116
# Feasible linear ILS times	36	36	27	33	30	30	25	217
# Feasible non-linear ILS times	36	28	16	20	29	24	21	174
# Feasible piecewise constant ILS pollution	35	27	19	16	10	6	6	119
# Feasible linear ILS pollution	35	27	20	16	10	6	6	120
Total generated solutions	36	36	27	36	33	36	36	240

Table 2: Feasibility results

We see that in total only 15 of the generated solutions are already feasible in the initial solution, which is about 6.3% of the generated initial solutions. Remarkable is that none of these feasible initial solutions is drawn from the large instances with more than 25 customers. After applying the ILS to these initial solutions, 48% of all solutions is feasible. Again, we see more feasibility for small instances than for large ones, as we see the number of feasible solutions drop when the number of customers increases.

When we only look at the feasibility in time for linear charging times, we see that 90.0% of the solutions are feasible in all time windows, and for the small instances with the number of customers less or equal to 15 all solutions are even feasible in the time windows. When we look at the direct effect of introducing the non-linear charging time, we see that less of our generated solutions are feasible, only 72.5%. We see that for $|\mathcal{N}| = 5$, the non-linear charging time has no impact on the feasibility of our instances. However, for instances with more customers, we see that there is an increasing impact as the instances get bigger. For example, when looking at the instances with $|\mathcal{N}| = 15$, the feasibility in time drops from 100% to only 59% when the linear charging time is assumed but non-linear charging time holds in reality.

When looking at the different functions for the pollution rate, we see that if the piecewise constant function is assumed, but the linear function holds, the solution remains feasible in the pollution rate in 100% of our cases. It even happens that one solution becomes feasible while it was not feasible with the piecewise constant function.

We now know how the ILS affects the overall feasibility from the initial to the best solution, and how the different functions for charging times and pollution affect the feasibility of a single component, so time and pollution, respectively. Table 3 provides some mean values to investigate how the solutions change numerically. Here, the mean values of the distances, times and pollution are given.

	$\lvert\mathcal{N}\rvert = 5$	$\lvert\mathcal{N}\rvert = 10$	$\lvert\mathcal{N}\rvert = 15$	$\lvert\mathcal{N}\rvert = 25$	$\lvert\mathcal{N}\rvert = 30$	$\lvert\mathcal{N}\rvert = 50$	$\lvert\mathcal{N}\rvert = 100$	Total
Initial distance	256.50	475.79	750.63	1022.30	1347.10	2156.19	4688.77	1559.60
ILS distance	202.31	342.94	440.42	480.19	597.73	1006.21	2372.21	790.78
Initial time	737.76	1471.69	2509.40	3556.64	4877.12	7627.95	18 025.61	5665.86
Linear ILS time	881.95	1739.21	2315.45	2349.87	2912.14	5087.33	10 304.07	3715.27
Non-linear ILS time	898.23	1772.40	2353.34	2369.90	2923.36	5099.64	10 319.07	3735.60
Initial pollution	86.13	193.02	319.43	390.34	472.20	902.78	2114.95	653.95
Piecewise constant ILS pollution	54.19	85.29	134.79	180.75	262.36	548.45	1478.43	403.30
Linear pollution	52.29	82.26	129.75	171.00	247.11	519.85	1396.21	381.86

Table 3: Mean results

We see that on average the distance is lowered by 49.3%, which is due to the ILS. Here we see that this percentage gets higher for larger sizes of the problem. For the instances of size $|\mathcal{N}| = 5$, the value is 21.1%, while for $|\mathcal{N}| = 100$, it is equal to 49.4%. We see a decrease in the mean time from the initial to the ILS solution. However, when we look at the solutions separately, it does not hold that all times decrease. Yet, by how the non-linear charging formula is generated, all non-linear times are greater or equal to the linear charging times. On average, the non-linear function takes about 0.5% extra time. This does not seem a lot, but as seen earlier in Table 2, this does have an impact on the feasibility of the time windows when the assumption of linear charging times is made, while the non-linear function holds.

For pollution, we see that the pollution value after the ILS is mostly lower than the pollution value in the initial solution. On average, this decrease is 38.3%. For the values $|\mathcal{N}| = 5, 10$ and 15, the decrease is higher for the larger problems. However, for $|\mathcal{N}| = 100$ the decrease is only 30.1%. When comparing the piecewise constant function for pollution against the linear function, we see that the linear value is always less or equal to the piecewise linear value. For our instances, it even turns out that the pollution is always lower for the linear function, except when no conventional vehicles are used and the pollution value is therefore 0.

To go even deeper into it, we look at some interesting instances for which $|\mathcal{N}| = 15$ and $\alpha = 0.75$, with corresponding Tables 4 and 5. In the first table, 'dist' denotes the distance, values with '(I)' are initial values, '(ILS)' is for the best solution found by the ILS assuming linear charging times, Time '(NL)' denotes the time the solution generated when assuming linear charging but non-linear charging holds and 'FNL' denotes if the time is feasible in this case. Furthermore, when the values of the second to the fifth column are boldface, it means that the solution is feasible. The way this table has been formatted has been done for all solutions found in the appendices.

For these instances, none of the initial solutions were feasible. We can see that the best solution found by the ILS in c106C15_0.75 and r105C15_0.75 is feasible. However, when using the non-linear time, they become infeasible. For the second instance, the increase in time is minimal, but enough to cause infeasibility.

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c106C15_0.75	695.3	**400.79**	3442.04	**3673.27**	3819.1	×
r105C15_0.75	709.21	**338.65**	1020.5	**626.35**	628.24	×
rc108C15_0.75	795.51	479.81	1015.77	704.22	716.78	✓

Table 4: Results for some instances with $|\mathcal{N}| = 15$ and $\alpha = 0.75$

In Table 5, 'FILS' denotes if the solution of the ILS is feasible in pollution, 'FLP' denotes if the solution is feasible in linear pollution, '(L)' is when linear pollution holds and 'Limit' denotes the pollution upper bound of the problem. The bolt values can only be found in the second and third columns here. For rc108C15_0.75, the solution of the ILS is overall infeasible, but also infeasible in pollution, as 228.99 is higher than the upper bound of 217.5. However, when using the linear pollution function, the pollution value drops to 215.65, which is lower than the upper bound, so the solution is feasible in pollution.

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c106C15_0.75	286.85	**15.4**	✓	14.92	✓	173.25
r105C15_0.75	327.81	**191.92**	✓	185.96	✓	207.75
rc108C15_0.75	477.61	228.99	×	215.65	✓	217.5

Table 5: Pollution results for some instances with $|\mathcal{N}| = 15$ and $\alpha = 0.75$

To further investigate the effect of the non-linear charging function, we look at the difference in the distance of the best solution when it is generated using the non-linear function compared to when it is generated using the linear charging function, as before. We can only compare distances when the non-linear charging time is feasible when the solution is generated while assuming that the function is linear. So there are two cases we compare. The case where non-linear charging is assumed and the case where linear charging is assumed, but non-linear charging holds. We can only compare these two when non-linear charging is feasible in the solution generated where we assume linear charging, such that we compare two feasible solutions with each other.

We compare these solutions for the small instances with $|\mathcal{N}| = 5$ and $|\mathcal{N}| = 10$ because for these instances the solution often remains feasible when the non-linear function holds. All results can be found in Appendix H and I, but some interesting results are given in Table 6. Here, the boldfaced values indicate whether the solution generated while assuming non-linear charging is better in terms of distance than the solution generated while assuming linear charging. All solutions are feasible, as we only compare solutions that are already feasible in case linear charging is assumed, but non-linear charging holds. In the table, the distance, time and piecewise constant pollution values are given for both the solutions generated while assuming linear charging (lin) and non-linear charging (non-lin). Lastly, the pollution upper bound is given, such that it is easy to verify that the solution is feasible in pollution and so that we can reason why a solution is better, for example by introducing more conventional routes if the pollution value is higher than before.

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c103C5_0.75	175.37	**165.67**	1767.31	**1764.25**	0.00	**57.02**	93.75
r202C5_0.75	143.04	143.04	228.91	228.91	53.77	53.77	81.75
r203C5_0.75	193.84	193.84	288.25	292.23	79.47	79.47	103.50

Table 6: Non-linear charging comparison for some interesting instances with $|\mathcal{N}| = 5$ and $\alpha = 0.75$

In this table, we see three different cases that are possible. In the first row, we see that the non-linear result is better than the linear result, but more pollution is used. This is fine because it is still below the pollution upper bound. In the second row, we see that the solution has not changed, not even if we look at the time. In the last row, we see that the solution remained the same, but the time differs.

For all instances with $|\mathcal{N}| = 5$, 20.5% of the solutions get better. We see that for the high pollution upper bound, $\alpha = 0.75$, 33.3% of the solutions get better, while for $\alpha = 0.50$ and $\alpha = 0.25$ only respectively 18.2% and 9.1%.

For instances with $|\mathcal{N}| = 10$, 57.1% of the solutions get better. When we break it into the three values of α, we see that 44.4%, 71.4% and 60% of the solutions get better for $\alpha = 0.75, 0.50$ and 0.25, respectively.

When looking at all better solutions, the distance is lowered by 14.1% on average for instances with 5 customers and 7.1% for instances with 10 customers.

7 Conclusion

In this study, we modelled and solved the Green Vehicle Routing Problem with Time Windows. We investigate the utility of using the ILS versus only the initial solution. Furthermore, we proposed two functions to test the effect of different functions, a non-linear charging function and a linear function for the pollution rate per km. We compared the results of the ILS against the initial results to investigate how good the initial solution is and what the utility of the ILS is. We found that the initial solution is not feasible in about 90% of the cases, which means the way the initial solution is generated is not of value to use in reality. However, after applying the ILS, 62% of the solutions are feasible, of which we can conclude that the ILS does its work by improving the solution, but not for every instance its solution is useful. When we take a closer look at some instances, we see that the ready time is sometimes equal to the due date, which makes it likely for the problem to become infeasible. Furthermore, the ILS ensures that the solution distance decreases by an average of 45.7%. From these two points, it can be concluded that the ILS is useful for improving the solution compared to the initial solution.

That time windows are not the only cause of infeasibility becomes clear from the fact that 90.0% of the time windows in the solutions are feasible. For the small instances, until 15 customers, all solutions are even feasible in their time window. When, however, when non-linear charging times are used on the linear solutions, only 72.5% of the solutions are feasible in their time window, which is quite an extreme loss. However, it is worth mentioning again that the non-linear function used in this study is generated with the assumption that the linear function is chosen optimistically. We can thus conclude that assuming linear charging times may, in the worst case, cause problems when non-linear holds.

When using the non-linear charging function is still feasible in the solution generated with the linear charging function, we can compare this solution to the solution we get when the non-linear is used when generating the solution. For instances with 5 customers, 20.5% of the solutions get better when assuming non-linear charging, with an average loss of 14.1% in distance. For these instances, we see that the percentage of better solutions is higher when α is higher, so when the upper bound is higher. For instances with 10 customers, even 57.1% of the solutions get better. However, here the average loss in distance is only 7.1%. Furthermore, for these instances, we do not see any pattern for the different values of the upper bound.

When using a linear expected pollution rate, we see that the values of pollution all go down compared to the piecewise constant function. From this, we can conclude that assuming the pollution rate to be piecewise constant does not cause infeasibility of the problem for our instances and when it is linear, it is only lower than expected, as a result of which the imposed condition will always be met, but the costs may be slightly higher than would have been possible.

For further research, the speed that is driven on each arc in the problem could be examined. In this study, an average speed has been taken, but in reality, this depends on the routes that are driven. As a result, the energy consumption per km will also depend on which routes are driven, which will change the use of charging stations. A different pollution rate will also arise for conventional cars, which depends on both the load and the speed. In addition, the costs of hiring drivers and the use of multiple vehicles can be taken into account.

References

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, *45*(8), 1232–1250.

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.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, *48*(1), 100–114.

Felipe, Á., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, *71*, 111–128.

Kancharla, S. R., & Ramadurai, G. (2020). Electric vehicle routing problem with non-linear charging and load-dependent discharging. *Expert Systems with Applications*, *160*, 113714.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, *70*, 239–254.

Koç, Ç., & Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, *39*, 154–164.

Macrina, G., Pugliese, L. D. P., 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.

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.

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

Sassi, O., Cherif, W. R., & Oulamara, A. (2014). 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*(4), 500–520.

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

Toth, P., & Vigo, D. (2002). *The vehicle routing problem.* SIAM.

Zhang, K., & Batterman, S. (2010). Near-road air pollutant concentrations of co and pm2. 5: A comparison of mobile6. 2/caline4 and generalized additive models. *Atmospheric Environment*, *44*(14), 1740–1748.

A Numerical results for $|\mathcal{N}| = 5$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C5_0.75	296.09	**234.82**	1417.70	**1623.73**	1623.73	✓
c103C5_0.75	**187.22**	**175.37**	**1043.99**	**1767.31**	1792.25	✓
c206C5_0.75	**313.92**	**233.94**	**1569.39**	**2031.03**	2066.95	✓
c208C5_0.75	282.82	**174.82**	891.08	**2029.29**	2062.26	✓
r104C5_0.75	201.46	**136.45**	386.3	**294.12**	294.12	✓
r105C5_0.75	211.3	**153.49**	300.97	**291.59**	291.59	✓
r202C5_0.75	190.04	**143.04**	530.27	**228.91**	235.88	✓
r203C5_0.75	268.38	**193.84**	374.03	**288.25**	292.23	✓
rc105C5_0.75	**262.95**	**233.9**	**373.04**	**338.23**	345.83	✓
rc108C5_0.75	301.46	**305.52**	394.46	**439.57**	452.98	✓
rc204C5_0.75	**284.44**	**176**	**925.66**	**915.24**	920.88	✓
rc208C5_0.75	290.42	**174.38**	648.27	**292.39**	300.1	✓

Table 7: Results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C5_0.75	78.96	**67.24**	✓	65.15	✓	135.75
c103C5_0.75	**53.9**	**0**	✓	0	✓	93.75
c206C5_0.75	**62.99**	**105.91**	✓	102.59	✓	126.75
c208C5_0.75	182.66	**0**	✓	0	✓	0.75
r104C5_0.75	84.53	**61.39**	✓	59.47	✓	81.75
r105C5_0.75	100.37	**78.41**	✓	75.97	✓	87.75
r202C5_0.75	87.63	**53.77**	✓	52.08	✓	81.75
r203C5_0.75	119.19	**79.47**	✓	76.98	✓	111.75
rc105C5_0.75	**126.68**	**127.05**	✓	122.28	✓	135
rc108C5_0.75	147.23	**69.3**	✓	67.13	✓	146.25
rc204C5_0.75	**61.68**	**65.06**	✓	63.02	✓	99.75
rc208C5_0.75	23.91	**0**	✓	0	✓	0.75

Table 8: Pollution results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C5_0.50	296.09	**234.82**	1417.7	**1623.73**	1623.73	✓
c103C5_0.50	**187.22**	**165.67**	**1043.99**	**1739.31**	1764.25	✓
c206C5_0.50	**313.92**	**238.67**	**1569.39**	**2420.98**	2458.66	✓
c208C5_0.50	282.82	157.72	891.08	1661.27	1661.27	✓
r104C5_0.50	**191.62**	**137.01**	**320.59**	**334.02**	339.04	✓
r105C5_0.50	211.3	**182.01**	300.97	**297.25**	305.66	✓
r202C5_0.50	192.2	**143.04**	279.55	**228.91**	235.88	✓
r203C5_0.50	283.08	**199.54**	388.73	**347.31**	367.01	✓
rc105C5_0.50	262.95	**237.76**	373.04	**341.42**	347.09	✓
rc108C5_0.50	301.46	**315.81**	394.46	**452.16**	457.35	✓
rc204C5_0.50	**284.44**	**176**	**925.66**	**915.24**	920.88	✓
rc208C5_0.50	290.42	**174.38**	648.27	**292.39**	300.1	✓

Table 9: Results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C5_0.50	78.96	**67.24**	✓	65.15	✓	90.5
c103C5_0.50	**53.9**	**57.02**	✓	55.25	✓	62.5
c206C5_0.50	**62.99**	**80**	✓	77.48	✓	84.5
c208C5_0.50	182.66	121.44	×	117.65	×	0.5
r104C5_0.50	**23.46**	**44.91**	✓	42.32	✓	54.5
r105C5_0.50	100.37	**47.84**	✓	46.34	✓	58.5
r202C5_0.50	89.29	**53.77**	✓	52.08	✓	54.5
r203C5_0.50	130.51	**0**	✓	0	✓	74.5
rc105C5_0.50	126.68	**77.98**	✓	74.73	✓	90
rc108C5_0.50	147.23	**87.24**	✓	81.74	✓	97.5
rc204C5_0.50	**61.68**	**65.06**	✓	63.02	✓	66.5
rc208C5_0.50	23.91	**0**	✓	0	✓	0.5

Table 10: Pollution results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C5_0.25	296.09	**278.47**	1624.04	**1556.62**	1749.84	✓
c103C5_0.25	187.22	**175.37**	1043.99	**1678.44**	1703.37	✓
c206C5_0.25	**299.57**	**242.56**	**1679.36**	**1823.14**	1971.17	✓
c208C5_0.25	282.82	**174.82**	891.08	**2029.29**	2062.26	✓
r104C5_0.25	**191.62**	**137.01**	**320.59**	**334.02**	339.04	✓
r105C5_0.25	211.3	**195.31**	300.97	**347.52**	357.85	✓
r202C5_0.25	170.85	**143.13**	269.77	**415.98**	415.98	✓
r203C5_0.25	241.5	**199.54**	660.51	**347.31**	367.01	✓
rc105C5_0.25	280.89	**239.46**	395.86	**382.85**	400.52	✓
rc108C5_0.25	307.76	337.48	400.75	488.45	493.98	✓
rc204C5_0.25	285.79	**185.16**	915.66	**307.37**	322.25	✓
rc208C5_0.25	290.42	**174.38**	648.27	**292.39**	300.1	✓

Table 11: Results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C5_0.25	33.17	**33.17**	✓	32.13	✓	45.25
c103C5_0.25	53.9	**19.72**	✓	19.1	✓	31.25
c206C5_0.25	**24.35**	**24.35**	✓	23.59	✓	42.25
c208C5_0.25	182.66	**0**	✓	0	✓	0.25
r104C5_0.25	**23.46**	**0**	✓	0	✓	27.25
r105C5_0.25	100.37	**0**	✓	0	✓	29.25
r202C5_0.25	54.68	**0**	✓	0	✓	27.25
r203C5_0.25	91.16	**0**	✓	0	✓	37.25
rc105C5_0.25	88.02	**0**	✓	0	✓	45
rc108C5_0.25	152.08	60.51	×	58.62	×	48.75
rc204C5_0.25	61.68	**0**	✓	0	✓	33.25
rc208C5_0.25	23.91	**0**	✓	0	✓	0.25

Table 12: Pollution results for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.25$

B Numerical results for $|\mathcal{N}| = 10$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C10_0.75	567.77	**486.27**	2685.27	**2928.9**	2928.9	×
c104C10_0.75	438.69	**317.55**	2072.33	**2207.55**	2400.07	✓
c202C10_0.75	443.34	**282.38**	2837.42	**5399.01**	5400.93	✓
c205C10_0.75	545.65	322.93	3668.74	4212.52	4212.52	✓
r102C10_0.75	429.84	**314.67**	808.53	**581.29**	583.45	✓
r103C10_0.75	**286.46**	**193.9**	**552.66**	**443.38**	443.38	✓
r201C10_0.75	**317.13**	**245.89**	**664.4**	**890.47**	896.24	✓
r203C10_0.75	397.97	**222.64**	1421.09	**704.68**	704.68	✓
rc102C10_0.75	589.17	**463.55**	794	**717.44**	737.44	×
rc108C10_0.75	531.9	**346.25**	677.16	**561.95**	573.83	✓
rc201C10_0.75	**487.21**	**416.77**	**1017.23**	**1164.19**	1165.71	✓
rc205C10_0.75	**538.95**	**349.16**	**1154.49**	**1093.69**	1093.69	✓

Table 13: Results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C10_0.75	170.18	**196.47**	✓	190.34	✓	222
c104C10_0.75	180.48	**24.83**	✓	24.05	✓	153.75
c202C10_0.75	171.62	**72.62**	✓	70.36	✓	138.75
c205C10_0.75	315.24	131.38	×	127.27	✓	131.25
r102C10_0.75	200.73	**106.24**	✓	100.84	✓	143.25
r103C10_0.75	**74.08**	**100.85**	✓	97.07	✓	113.25
r201C10_0.75	**113.3**	**90.38**	✓	87.54	✓	118.5
r203C10_0.75	177.21	**0**	✓	0	✓	126
rc102C10_0.75	293.97	**223.27**	✓	214.77	✓	227.25
rc108C10_0.75	327.02	**172.1**	✓	163.87	✓	195.75
rc201C10_0.75	**154.54**	**95.9**	✓	92.89	✓	176.25
rc205C10_0.75	**175.26**	**116.54**	✓	112.89	✓	190.5

Table 14: Pollution results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C10_0.50	567.77	366.33	2685.27	2609.19	2609.19	✓
c104C10_0.50	438.69	**302.68**	2072.33	**2252.98**	2487.23	×
c202C10_0.50	443.34	**282.38**	2837.42	**5399.01**	5400.93	✓
c205C10_0.50	550.3	425.93	2776.35	4570.38	4614.78	✓
r102C10_0.50	429.84	**319.06**	808.53	**580.2**	580.99	✓
r103C10_0.50	**286.46**	**199.31**	**552.66**	**475.74**	475.74	✓
r201C10_0.50	317.13	**235.13**	664.4	**1044.38**	1044.38	✓
r203C10_0.50	420.32	**233.21**	842.12	**757.8**	758.27	✓
rc102C10_0.50	589.17	**484.19**	794	**754.76**	795.76	×
rc108C10_0.50	531.9	437.15	677.16	642.43	660.76	✓
rc201C10_0.50	487.21	**360.35**	1017.23	**1204.13**	1211.73	✓
rc205C10_0.50	579.34	**351.2**	1091.74	**1132.33**	1132.33	✓

Table 15: Results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C10_0.50	170.18	289.21	×	273.36	×	148
c104C10_0.50	180.48	**0**	✓	0	✓	102.5
c202C10_0.50	171.62	**72.62**	✓	70.36	✓	92.5
c205C10_0.50	318.83	124.88	×	120.97	×	87.5
r102C10_0.50	200.73	**49.99**	✓	48.43	✓	95.5
r103C10_0.50	**74.08**	**48.91**	✓	47.38	✓	75.5
r201C10_0.50	113.3	**73.25**	✓	70.96	✓	79
r203C10_0.50	194.43	**0**	✓	0	✓	84
rc102C10_0.50	293.97	**50.5**	✓	48.92	✓	151.5
rc108C10_0.50	327.02	132.37	×	124.77	✓	130.5
rc201C10_0.50	154.54	**116.38**	✓	112.73	✓	117.5
rc205C10_0.50	206.36	**0**	✓	0	✓	127

Table 16: Pollution results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C10_0.25	605.76	482.6	2722.61	3073.52	3180.25	×
c104C10_0.25	508.02	**307.48**	2141.65	**2274.41**	2541.56	×
c202C10_0.25	445.27	283.59	3731.22	3931.79	3931.79	✓
c205C10_0.25	486.44	**267.06**	2109.43	**3939.67**	4022.46	×
r102C10_0.25	401.45	241.14	705.2	555.19	555.19	✓
r103C10_0.25	304.93	**216.52**	571.13	**448.1**	456.12	✓
r201C10_0.25	353.13	**278.55**	742.4	**1010.72**	1010.72	✓
r203C10_0.25	506.97	**222.64**	1408.53	**704.68**	704.68	✓
rc102C10_0.25	569.73	384.82	763.74	557.29	557.29	✓
rc108C10_0.25	558.36	510.01	739.61	754.47	773.15	✓
rc201C10_0.25	492.8	**362.01**	1016.26	**1548.21**	1568.22	✓
rc205C10_0.25	636.3	**349.18**	1241.75	**1105.8**	1105.8	✓

Table 17: Results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C10_0.25	196.41	107.95	×	102.82	×	74
c104C10_0.25	228.38	**0**	✓	0	✓	51.25
c202C10_0.25	173.11	65.56	×	63.51	×	46.25
c205C10_0.25	214.13	**0**	✓	0	✓	43.75
r102C10_0.25	181.5	189.08	×	179.93	×	47.75
r103C10_0.25	85.09	**18.26**	✓	17.69	✓	37.75
r201C10_0.25	141.03	**0**	✓	0	✓	39.5
r203C10_0.25	50.48	**0**	✓	0	✓	42
rc102C10_0.25	276.98	300.06	×	287.13	×	75.75
rc108C10_0.25	342.83	107.48	×	104.13	×	65.25
rc201C10_0.25	158.85	**0**	✓	0	✓	58.75
rc205C10_0.25	158.44	**0**	✓	0	✓	63.5

Table 18: Pollution results for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.25$

C Numerical results for $|\mathcal{N}| = 15$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c106C15_0.75	695.3	**400.79**	3442.04	**3673.27**	3819.1	×
c202C15_0.75	872.84	514.12	5607.33	6355.72	6440.17	✓
c208C15_0.75	736.38	**306.25**	5320.55	**3437.54**	3437.54	✓
r105C15_0.75	709.21	**338.65**	1020.5	**626.35**	628.24	×
r202C15_0.75	646.96	**491.62**	2374.31	**1903.4**	1907.65	✓
rc103C15_0.75	691.32	**515.15**	1051.62	**810.22**	813.69	✓
rc108C15_0.75	795.51	479.81	1015.77	704.22	716.78	✓
rc202C15_0.75	730.69	**513.54**	1828.17	**1844.32**	1844.32	✓
rc204C15_0.75	668.78	**326.22**	1506.09	**1173.81**	1175.18	✓

Table 19: Results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c106C15_0.75	286.85	**15.4**	✓	14.92	✓	173.25
c202C15_0.75	276.27	225.72	×	218.66	×	213.75
c208C15_0.75	218.51	**24.35**	✓	23.59	✓	171.75
r105C15_0.75	327.81	**191.92**	✓	185.96	✓	207.75
r202C15_0.75	278.87	**112.47**	✓	108.95	✓	118.5
rc103C15_0.75	421.72	**211.34**	✓	202.05	✓	226.5
rc108C15_0.75	477.61	228.99	×	215.65	✓	217.5
rc202C15_0.75	238.94	**71.68**	✓	69.43	✓	235.5
rc204C15_0.75	245.67	**86.05**	✓	83.36	✓	176.25

Table 20: Pollution results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c106C15_0.50	695.3	**493.57**	3442.04	**3851.42**	4015.74	✓
c202C15_0.50	933.02	**434.27**	6513.26	**5929.23**	5929.23	×
c208C15_0.50	817.65	759.91	5617.41	4754.97	4856.17	×
r105C15_0.50	709.21	394.72	1020.5	662.23	677.19	×
r202C15_0.50	766.64	**442.75**	2109.85	**1698.35**	1701.95	✓
rc103C15_0.50	691.32	545.47	1051.62	856.72	860.74	✓
rc108C15_0.50	795.51	**481.13**	1015.77	**761.68**	792.27	×
rc202C15_0.50	730.69	**513.79**	1828.17	**2096.1**	2108.03	✓
rc204C15_0.50	636.84	**326.22**	1566.37	**1173.81**	1175.18	✓

Table 21: Results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c106C15_0.50	286.85	**63.63**	✓	61.64	✓	115.5
c202C15_0.50	322.61	**91.55**	✓	88.68	✓	142.5
c208C15_0.50	281.1	213.75	×	207.05	×	114.5
r105C15_0.50	327.81	168.23	×	162.99	×	138.5
r202C15_0.50	371.03	**70.92**	✓	68.7	✓	79
rc103C15_0.50	421.72	201.12	×	193.71	×	151
rc108C15_0.50	477.61	**117.56**	✓	109.28	✓	145
rc202C15_0.50	238.94	**122.25**	✓	118.41	✓	157
rc204C15_0.50	221.08	**86.05**	✓	83.36	✓	117.5

Table 22: Pollution results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c106C15_0.25	722.92	466.86	3282.94	3854.07	4018.39	✓
c202C15_0.25	875.37	553.24	4364.42	7668.28	7671.27	✓
c208C15_0.25	805.22	413.03	4965.23	5241.58	5241.58	✓
r105C15_0.25	709.21	342.15	1020.5	617.28	632.24	×
r202C15_0.25	887.68	356.09	1803.91	1309.27	1309.27	✓
rc103C15_0.25	670.46	844.99	944.33	1238.4	1250.44	✓
rc108C15_0.25	784.94	506.94	1005.2	782.54	803.99	✓
rc202C15_0.25	737.23	529.02	1610.77	2223.93	2244.54	✓
rc204C15_0.25	764.6	**354.96**	1445.37	**1152.17**	1153.54	✓

Table 23: Results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c106C15_0.25	308.11	70.24	×	68.04	×	57.75
c202C15_0.25	278.22	141.27	×	136.84	×	71.25
c208C15_0.25	271.52	144.91	×	140.38	×	57.25
r105C15_0.25	327.81	180.17	×	174.56	×	69.25
r202C15_0.25	277.66	274.19	×	265.63	×	39.5
rc103C15_0.25	405.66	170.06	×	164.73	×	75.5
rc108C15_0.25	465.68	175.46	×	168.07	×	72.5
rc202C15_0.25	243.98	118.87	×	115.15	×	78.5
rc204C15_0.25	319.46	**0**	✓	0	✓	58.75

Table 24: Pollution results for instances with $|\mathcal{N}| = 15$ and $\alpha = 0.25$

D Numerical results for $|\mathcal{N}| = 25$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C25_0.75	1376.08	707.2	5829.02	3653.65	3653.65	×
c103C25_0.75	851.15	333.59	6241.39	2668.25	2668.25	✓
c206C25_0.75	1092.65	**384.48**	6906.57	**3443.79**	3484.83	×
c208C25_0.75	1146.04	**272.57**	6414.94	**3215.4**	3325.06	×
r104C25_0.75	751.54	**366.98**	1352.6	**824.81**	841.79	×
r105C25_0.75	979.19	515.98	1544.57	1056.07	1058.27	×
r202C25_0.75	**789**	**404.37**	**2545.36**	**2815.79**	2815.79	✓
r203C25_0.75	804.97	**418.33**	4007.45	**2822.28**	2822.28	✓
rc105C25_0.75	1156.46	658.68	1814.49	1296.46	1296.46	×
rc108C25_0.75	945.95	530.15	1371.55	950.62	953.9	✓
rc204C25_0.75	1003.8	**376.33**	2565.61	**2029.28**	2029.28	✓
rc208C25_0.75	816.53	**371.78**	1645.77	**1063.73**	1063.73	✓

Table 25: Results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C25_0.75	672.37	566.61	×	527.78	×	418.5
c103C25_0.75	241.84	274.39	×	249.02	×	206.25
c206C25_0.75	349.66	**91.29**	✓	86.36	✓	146.25
c208C25_0.75	390.77	**0**	✓	0	✓	146.25
r104C25_0.75	298	**148.54**	✓	140.65	✓	176.25
r105C25_0.75	325.36	193.05	×	184.93	×	153.75
r202C25_0.75	**211.3**	**93.76**	✓	90.81	✓	228.75
r203C25_0.75	277.85	**125.66**	✓	121.72	✓	195.75
rc105C25_0.75	660.53	406.34	×	386	×	352.5
rc108C25_0.75	536.23	271.49	×	253.3	×	235.5
rc204C25_0.75	418.96	**0**	✓	0	✓	75
rc208C25_0.75	314.14	**0**	✓	0	✓	202.5

Table 26: Pollution results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C25_0.50	1376.08	702.84	5829.02	4103.75	4103.75	×
c103C25_0.50	991.51	544.74	6660.79	4406.97	4513.75	✓
c206C25_0.50	1242.7	554.05	6775.5	4705.03	4705.03	×
c208C25_0.50	1146.04	460.48	6414.94	3953.76	4063.02	×
r104C25_0.50	769.31	343.83	1344.13	767.39	771.85	✓
r105C25_0.50	979.19	604.08	1544.57	1165.22	1178.18	✓
r202C25_0.50	789	**421.63**	2545.36	**3360.12**	3360.12	✓
r203C25_0.50	804.97	**360.65**	4007.45	**1784.06**	1784.06	✓
rc105C25_0.50	1156.46	676.52	1814.49	1330.87	1330.87	×
rc108C25_0.50	945.95	466.97	1371.55	859.98	859.98	✓
rc204C25_0.50	967.31	**381.21**	2453.35	**1519.71**	1519.71	✓
rc208C25_0.50	800.55	**401.36**	1330.27	**1249.79**	1249.79	✓

Table 27: Results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C25_0.50	672.37	554.24	×	524.48	×	279
c103C25_0.50	351.78	199.4	×	189.34	×	137.5
c206C25_0.50	465.19	129.63	×	125.58	×	97.5
c208C25_0.50	390.77	152.44	×	147.68	×	97.5
r104C25_0.50	309.43	138.91	×	132.2	×	117.5
r105C25_0.50	325.36	195.13	×	186.93	×	102.5
r202C25_0.50	211.3	**93.76**	✓	90.81	✓	152.5
r203C25_0.50	277.85	**77.5**	✓	75.07	✓	130.5
rc105C25_0.50	660.53	421.43	×	400.36	×	235
rc108C25_0.50	536.23	267.21	×	247.1	×	157
rc204C25_0.50	237.7	**37.65**	✓	36.47	✓	50
rc208C25_0.50	301.84	**65.69**	✓	63.63	✓	135

Table 28: Pollution results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C25_0.25	1376.08	704.77	5829.02	3523.67	3523.67	×
c103C25_0.25	1210.62	498.99	7364.85	4401.26	4503.54	×
c206C25_0.25	1456.32	**653.94**	7318.07	**5408.92**	5437.01	×
c208C25_0.25	1402.78	**489.6**	7456.36	**4121.99**	4276.3	×
r104C25_0.25	810.79	423.69	1407.69	987.85	998.87	✓
r105C25_0.25	1030.05	530.15	1572.62	1158.61	1168.57	×
r202C25_0.25	927.88	460.52	2472.35	2496.19	2496.19	✓
r203C25_0.25	783.88	**361.71**	3160.82	**2557.01**	2557.01	✓
rc105C25_0.25	1156.46	644.18	1814.49	1211.6	1220.22	×
rc108C25_0.25	928.86	412.16	1374.34	775.27	775.27	✓
rc204C25_0.25	1022.5	**405.84**	2230.71	**1633.23**	1633.23	✓
rc208C25_0.25	1014.13	**442.46**	1706.86	**1273.14**	1273.14	✓

Table 29: Results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C25_0.25	672.37	556.43	×	525.93	×	139.5
c103C25_0.25	508.65	255.29	×	241.44	×	68.75
c206C25_0.25	274.08	**0**	✓	0	✓	48.75
c208C25_0.25	219.63	**47.44**	✓	45.95	✓	48.75
r104C25_0.25	339.31	165.91	×	158.53	×	58.75
r105C25_0.25	364.51	126.51	×	120.96	×	51.25
r202C25_0.25	318.24	88.22	×	85.45	×	76.25
r203C25_0.25	261.61	**50.14**	✓	48.57	✓	65.25
rc105C25_0.25	660.53	344.75	×	328.62	×	117.5
rc108C25_0.25	521.34	334.59	×	307.63	×	78.5
rc204C25_0.25	127.86	**0**	✓	0	✓	25
rc208C25_0.25	346.68	**33.57**	✓	32.51	✓	67.5

Table 30: Pollution results for instances with $|\mathcal{N}| = 25$ and $\alpha = 0.25$

E Numerical results for $|\mathcal{N}| = 30$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C30_0.75	1696.99	897.71	6830.62	4228.23	4228.23	×
c103C30_0.75	1382.76	424.56	8720.29	3668.69	3668.69	✓
c206C30_0.75	1536.11	537.17	10 286.64	4928.6	4961.13	✓
c208C30_0.75	1446.55	685.04	8950.66	5083.93	5108.33	✓
r105C30_0.75	1189.19	674.89	1875.18	1256.17	1268.22	✓
r202C30_0.75	1108.53	**591.42**	4484.91	**3654.04**	3654.04	✓
r203C30_0.75	978.35	**409.73**	2771.49	**2539.63**	2539.63	✓
rc105C30_0.75	1486.31	727.75	2109.08	1239.17	1239.17	✓
rc108C30_0.75	1302.3	514.29	1843.38	858.2	858.2	✓
rc204C30_0.75	1278.73	**413.06**	2840.62	**1867.48**	1867.48	✓
rc208C30_0.75	1207.09	**486.1**	1988.73	**1205.63**	1209.06	✓

Table 31: Results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C30_0.75	832.09	716.17	×	669.91	×	543
c103C30_0.75	566.51	349.22	×	316.95	×	277.5
c206C30_0.75	592.84	229.45	×	220.74	×	162
c208C30_0.75	537.42	210.34	×	203.76	×	165.75
r105C30_0.75	438.1	312.96	×	292.14	×	279.75
r202C30_0.75	312.27	**115.95**	✓	112.31	✓	250.5
r203C30_0.75	281.58	**37.61**	✓	36.43	✓	211.5
rc105C30_0.75	789.92	476.31	×	453.16	×	433.5
rc108C30_0.75	677.89	413.27	×	383.82	×	324.75
rc204C30_0.75	59.24	**0**	✓	0	✓	0.75
rc208C30_0.75	478.92	**96.24**	✓	93.22	✓	100.5

Table 32: Pollution results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C30_0.50	1696.99	873.38	6830.62	4830.98	4830.98	×
c103C30_0.50	1382.76	423.3	8720.29	3347.95	3347.95	✓
c206C30_0.50	1421.99	564.99	9522.52	4812.73	4841	✓
c208C30_0.50	1404.39	528.31	9437.61	4426.19	4446.69	✓
r105C30_0.50	1189.19	701.97	1875.18	1406.16	1411.1	✓
r202C30_0.50	1108.53	**640.15**	4484.91	**3996.52**	3996.52	✓
r203C30_0.50	978.35	**639.77**	2771.49	**2386.96**	2386.96	✓
rc105C30_0.50	1486.31	727.75	2109.08	1239.17	1239.17	✓
rc108C30_0.50	1302.3	494.98	1843.38	888.57	888.57	✓
rc204C30_0.50	1278.73	**434.49**	2840.62	**1937.65**	1937.65	✓
rc208C30_0.50	1281.36	628.77	2568.67	1531.09	1531.09	✓

Table 33: Results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C30_0.50	832.09	694.29	×	651.74	×	362
c103C30_0.50	566.51	349.03	×	315.98	×	185
c206C30_0.50	504.96	230.28	×	223.09	×	108
c208C30_0.50	504.96	152.95	×	145.51	×	110.5
r105C30_0.50	438.1	317.95	×	295.13	×	186.5
r202C30_0.50	312.27	**110.19**	✓	106.73	✓	167
r203C30_0.50	281.58	**109.83**	✓	106.38	✓	141
rc105C30_0.50	789.92	476.31	×	453.16	×	289
rc108C30_0.50	677.89	395.46	×	369.4	×	216.5
rc204C30_0.50	59.24	**0**	✓	0	✓	0.5
rc208C30_0.50	422.78	111.19	×	107.71	×	67

Table 34: Pollution results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C30_0.25	1696.99	898.02	6830.62	5257.96	5257.96	×
c103C30_0.25	1470.25	420.29	8405.12	3436.68	3436.68	✓
c206C30_0.25	1730.84	611.42	10 960.19	6193.19	6299.05	✓
c208C30_0.25	1730.59	**714.82**	9758.73	**6555.65**	6660.21	×
r105C30_0.25	1243.04	687.83	1869.81	1215.37	1234.33	✓
r202C30_0.25	1136.42	458.09	4555.32	2897.97	2897.97	✓
r203C30_0.25	1021.37	**451.35**	3505.72	**3054.59**	3054.59	✓
rc105C30_0.25	1486.31	802.46	2109.08	1462.48	1468.02	✓
rc108C30_0.25	1259.37	568.89	1784.76	947.5	956.72	✓
rc204C30_0.25	1278.73	**476.09**	2840.62	**1866.93**	1866.93	✓
rc208C30_0.25	1256.54	616.26	2619.01	1878.59	1878.59	✓

Table 35: Results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C30_0.25	832.09	713.45	×	670.15	×	181
c103C30_0.25	635.72	342.8	×	313.74	×	92.5
c206C30_0.25	186.03	66.63	×	64.56	×	54
c208C30_0.25	186.03	**53.73**	✓	52.05	✓	55.25
r105C30_0.25	478	288.13	×	271.54	×	93.25
r202C30_0.25	333.75	352.73	×	341.69	×	83.5
r203C30_0.25	314.7	**28.43**	✓	27.53	✓	70.5
rc105C30_0.25	789.92	491.19	×	466.19	×	144.5
rc108C30_0.25	640.09	375.9	×	351.41	×	108.25
rc204C30_0.25	59.24	**0**	✓	0	✓	0.25
rc208C30_0.25	169.95	39.79	×	38.54	×	33.5

Table 36: Pollution results for instances with $|\mathcal{N}| = 30$ and $\alpha = 0.25$

F Numerical results for $|\mathcal{N}| = 50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C50_0.75	2742.53	1773.6	12 454.24	8063.44	8063.44	×
c103C50_0.75	2411.48	**911.62**	11 487.44	**8679.45**	8775.52	×
c206C50_0.75	2569.95	1125.17	17 117.71	13 167.15	13 184.58	✓
c208C50_0.75	2463.31	1188.69	16 183.59	12 291.89	12 326.88	✓
r104C50_0.75	1772.72	713.14	2905.13	1424.7	1424.7	✓
r105C50_0.75	2194.73	1431.13	3153.99	2477.62	2481.44	×
r202C50_0.75	1760.23	**735.38**	8901.96	**5825.15**	5825.15	✓
r203C50_0.75	1636.04	**677.34**	4960.25	**3961.86**	3961.86	✓
rc105C50_0.75	2199.04	1572.06	3350.57	2200.5	2200.5	×
rc108C50_0.75	2119.78	828.06	3075.27	1538.14	1545.02	✓
rc204C50_0.75	1928.58	**602.04**	4874.62	**2831.56**	2843.66	✓
rc208C50_0.75	1874.31	**637.25**	3320.06	**1970.87**	1970.87	✓

Table 37: Results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C50_0.75	1530.06	1416.34	×	1323.64	×	1085.25
c103C50_0.75	1322.08	**547.14**	✓	508.59	✓	877.5
c206C50_0.75	911.4	436.86	×	423.22	×	291
c208C50_0.75	819.35	478.98	×	464.02	×	161.25
r104C50_0.75	895.27	565.82	×	532.17	×	432
r105C50_0.75	1121.54	717.72	×	686.64	×	432.75
r202C50_0.75	532.09	**205.93**	✓	199.51	✓	235.5
r203C50_0.75	461.57	**103.56**	✓	100.31	✓	345.75
rc105C50_0.75	1006.57	1235.43	×	1173.08	×	882
rc108C50_0.75	1016.64	547.13	×	508.24	×	282.75
rc204C50_0.75	629.6	**210.07**	✓	203.5	✓	214.5
rc208C50_0.75	599.2	**177.66**	✓	172.1	✓	201

Table 38: Pollution results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C50_0.50	2742.53	1799.77	12 454.24	7677.43	7677.43	×
c103C50_0.50	2411.48	1265.14	11 487.44	10 771.5	10 824.42	×
c206C50_0.50	2569.95	830.19	17 117.71	9435.14	9497.17	✓
c208C50_0.50	2583.57	1175.92	16 977.41	10 874.09	10 909.07	✓
r104C50_0.50	1772.72	845.59	2905.13	1622.32	1622.32	✓
r105C50_0.50	2194.73	1431.13	3153.99	2477.62	2481.44	×
r202C50_0.50	1760.23	805.54	8901.96	6600.96	6600.96	✓
r203C50_0.50	1636.04	637.76	4960.25	4039.46	4039.46	✓
rc105C50_0.50	2199.04	1457.35	3350.57	2064.01	2064.01	×
rc108C50_0.50	2119.78	755.71	3075.27	1420.78	1420.78	✓
rc204C50_0.50	1890.08	744.16	3960.13	3046.91	3053.82	✓
rc208C50_0.50	1979.88	702.51	3903.48	2316.37	2316.37	✓

Table 39: Results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C50_0.50	1530.06	1424.32	×	1343.03	×	723.5
c103C50_0.50	1322.08	674.65	×	632.23	×	585
c206C50_0.50	911.4	331.49	×	319.55	×	194
c208C50_0.50	911.94	475.78	×	460.93	×	107.5
r104C50_0.50	895.27	668.36	×	630.97	×	288
r105C50_0.50	1121.54	717.72	×	686.64	×	288.5
r202C50_0.50	532.09	208.16	×	201.66	×	157
r203C50_0.50	461.57	491.08	×	475.74	×	230.5
rc105C50_0.50	1006.57	1154.7	×	1087.47	×	588
rc108C50_0.50	1016.64	493.4	×	453.74	×	188.5
rc204C50_0.50	599.96	149.61	×	144.92	×	143
rc208C50_0.50	680.49	237.62	×	230.18	×	134

Table 40: Pollution results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101C50_0.25	2742.53	1814.66	12 454.24	8409.24	8409.24	×
c103C50_0.25	2411.48	828.66	11 487.44	8318.41	8425.69	×
c206C50_0.25	2473.86	646.7	16 594.59	8806.47	8806.47	✓
c208C50_0.25	2670.82	854.33	16 330.97	7730.48	7730.48	✓
r104C50_0.25	1772.72	784.68	2905.13	1516.25	1516.25	✓
r105C50_0.25	2194.73	1431.13	3153.99	2477.62	2481.44	×
r202C50_0.25	1890.75	820.89	7511.23	6341.78	6341.78	✓
r203C50_0.25	1667.25	698.84	5579.31	3952.68	3952.68	✓
rc105C50_0.25	2199.04	1570.72	3350.57	2310.96	2310.96	×
rc108C50_0.25	2089.04	727.32	2961.15	1463.79	1463.79	✓
rc204C50_0.25	2033.6	**636.98**	4848.19	**2254.46**	2254.46	✓
rc208C50_0.25	1944.16	762.5	3396.91	2782.79	2782.79	✓

Table 41: Results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101C50_0.25	1530.06	1437.37	×	1354.21	×	361.75
c103C50_0.25	1322.08	462.25	×	427.66	×	292.5
c206C50_0.25	837.41	261.87	×	251.82	×	97
c208C50_0.25	491.93	223.76	×	214.43	×	53.75
r104C50_0.25	895.27	621.74	×	585.51	×	144
r105C50_0.25	1121.54	717.72	×	686.64	×	144.25
r202C50_0.25	632.59	186.4	×	180.55	×	78.5
r203C50_0.25	485.61	175.72	×	170.23	×	115.25
rc105C50_0.25	1006.57	1235.66	×	1172.07	×	294
rc108C50_0.25	988.23	579.99	×	542.76	×	94.25
rc204C50_0.25	707.26	**43.35**	✓	41.99	✓	71.5
rc208C50_0.25	646.42	128.73	×	124.7	×	67

Table 42: Pollution results for instances with $|\mathcal{N}| = 50$ and $\alpha = 0.25$

G Numerical results for $|\mathcal{N}| = 100$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101_21_0.75	5159.31	3802.95	23 615.36	16 445.07	16 445.07	×
c103_21_0.75	4263.76	3814.14	22 992.05	20 406.52	20 406.52	×
c206_21_0.75	5044.78	1394.89	51 165.53	18 138.47	18 201.49	×
c208_21_0.75	5008.39	**1482.41**	46 350.29	**19209.52**	19 351.48	✓
r104_21_0.75	4060.36	3304.22	6217.58	4666.39	4666.39	×
r105_21_0.75	4676.68	**2348.54**	6918.68	**4573.31**	4602.76	×
r202_21_0.75	4126.77	1639.27	14 511.7	8930.56	8930.56	✓
r203_21_0.75	3802.55	1796.76	13 035.88	9253.74	9253.74	✓
rc105_21_0.75	6021.51	4389.36	7900.49	5610.88	5610.88	×
rc108_21_0.75	5273.5	2518.28	6943.97	3857.72	3878.71	✓
rc204_21_0.75	4325.76	**1226.6**	9808.73	**5422.55**	5426.1	✓
rc208_21_0.75	4507.52	1529.16	8508.86	4386.59	4393.77	✓

Table 43: Results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.75$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101_21_0.75	2873.29	3079.05	×	2838.67	×	2261.25
c103_21_0.75	2618.39	3032.07	×	2846.39	×	2253
c206_21_0.75	1871.92	697.5	×	664.58	✓	684
c208_21_0.75	1900.23	**771.84**	✓	740.73	✓	824.25
r104_21_0.75	1734.61	2601.35	×	2465.59	×	1917.75
r105_21_0.75	2427.72	**1268.2**	✓	1207.02	✓	1926
r202_21_0.75	1369.94	454.78	×	440.54	×	415.5
r203_21_0.75	1161.65	566.35	×	548.63	×	379.5
rc105_21_0.75	3360.27	3483.08	×	3275.57	×	2079.75
rc108_21_0.75	2859.83	1722.18	×	1619.41	×	1558.5
rc204_21_0.75	1601.94	**642.66**	✓	612.86	✓	677.25
rc208_21_0.75	1605.37	571.66	×	552.91	×	543

Table 44: Pollution results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.75$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101_21_0.50	5159.31	3785.13	23 615.36	16 062.81	16 062.81	×
c103_21_0.50	4263.76	3731.45	22 992.05	18 480.81	18 480.81	×
c206_21_0.50	5044.78	2097.92	51 165.53	23 353.32	23 431.44	×
c208_21_0.50	5008.39	2251.61	46 350.29	26 507.13	26 514.41	✓
r104_21_0.50	4060.36	3313.14	6217.58	4756.61	4756.61	×
r105_21_0.50	4676.68	2882.96	6918.68	4836.19	4866.8	✓
r202_21_0.50	4126.77	1464.09	14 511.7	7509.13	7509.13	✓
r203_21_0.50	3802.55	1796.76	13 035.88	9253.74	9253.74	✓
rc105_21_0.50	6021.51	4407.59	7900.49	6051.91	6103.3	×
rc108_21_0.50	5273.5	1569.15	6943.97	2704.38	2704.38	✓
rc204_21_0.50	4325.76	1499.67	9808.73	5893.36	5893.36	×
rc208_21_0.50	4507.52	1368.7	8508.86	4189.35	4189.35	✓

Table 45: Results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.50$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101_21_0.50	2873.29	3014.67	×	2824.9	×	1507.5
c103_21_0.50	2618.39	2963.42	×	2784.7	×	1502
c206_21_0.50	1871.92	1043.11	×	1004.35	×	456
c208_21_0.50	1900.23	1051.95	×	1013.28	×	549.5
r104_21_0.50	1734.61	2616.17	×	2472.3	×	1278.5
r105_21_0.50	2427.72	1501.85	×	1431.41	×	1284
r202_21_0.50	1369.94	357.47	×	346.28	×	277
r203_21_0.50	1161.65	566.35	×	548.63	×	253
rc105_21_0.50	3360.27	2676.31	×	2539.15	×	1386.5
rc108_21_0.50	2859.83	1234.84	×	1132.09	×	1039
rc204_21_0.50	1601.94	792.39	×	758.48	×	451.5
rc208_21_0.50	1605.37	456.57	×	442.31	×	362

Table 46: Pollution results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.50$

Instance	Dist (I)	Dist (ILS)	Time (I)	Time (ILS)	Time (NL)	FNL
c101_21_0.25	5159.31	3716.8	23 615.36	17 127.28	17 127.28	×
c103_21_0.25	4263.76	3774.81	22 992.05	18 714.7	18 714.7	×
c206_21_0.25	5116.98	2163.06	48 438.08	21 940.6	22 010.95	×
c208_21_0.25	5008.39	1519.17	46 350.29	19 248.77	19 390.73	✓
r104_21_0.25	4060.36	3295.67	6217.58	4624.06	4624.06	×
r105_21_0.25	4676.68	2327.09	6918.68	4091.5	4120.2	✓
r202_21_0.25	4126.77	1625.06	14 511.7	8575.39	8575.39	✓
r203_21_0.25	3952	1144.52	13 403.52	6861.96	6861.96	✓
rc105_21_0.25	6021.51	4499.5	7900.49	6099.25	6163.98	×
rc108_21_0.25	5273.5	1733.99	6943.97	3017.52	3017.52	✓
rc204_21_0.25	4597.61	1011.99	9568.9	5133.82	5133.82	✓
rc208_21_0.25	4880.39	1462.78	9261.68	4344.54	4357.19	✓

Table 47: Results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.25$

Instance	Pollution (I)	Pollution (ILS)	FILS	Pollution (L)	FLP	Limit
c101_21_0.25	2873.29	2934.05	×	2773.61	×	753.75
c103_21_0.25	2618.39	3003.39	×	2817.14	×	751
c206_21_0.25	1927.52	1121.44	×	1073.45	×	228
c208_21_0.25	1900.23	799.66	×	768.14	×	274.75
r104_21_0.25	1734.61	2584.79	×	2459.08	×	639.25
r105_21_0.25	2427.72	1252.81	×	1186.18	×	642
r202_21_0.25	1369.94	446.28	×	432.31	×	138.5
r203_21_0.25	1275.86	490.96	×	475.6	×	126.5
rc105_21_0.25	3360.27	2946.81	×	2725.76	×	693.25
rc108_21_0.25	2859.83	1306.12	×	1207.51	×	519.5
rc204_21_0.25	1806.04	790.82	×	755.05	×	225.75
rc208_21_0.25	1892.48	515.4	×	499.29	×	181

Table 48: Pollution results for instances with $|\mathcal{N}| = 100$ and $\alpha = 0.25$

H Comparison results for non-linear charging for $|\mathcal{N}| = 5$

In this and the next section, a value in bold represents that a better solution, in terms of distance, is found for non-linear than for linear charging times. All "linear charged" solutions in this section were feasible and also feasible in non-linear charging.

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c101C5_0.75	234.82	**234.72**	1623.73	**1480.66**	67.24	**130.73**	135.75
c103C5_0.75	175.37	**165.67**	1767.31	**1764.25**	0.00	**57.02**	93.75
c206C5_0.75	233.94	**167.99**	2031.03	**2102.83**	105.91	**73.49**	126.75
c208C5_0.75	174.82	174.82	2029.29	2048.19	0.00	0.00	0.75
r104C5_0.75	136.45	136.45	294.12	294.12	61.39	61.39	81.75
r105C5_0.75	153.49	153.49	291.59	292.21	78.41	78.41	87.75
r202C5_0.75	143.04	143.04	228.91	228.91	53.77	53.77	81.75
r203C5_0.75	193.84	193.84	288.25	292.23	79.47	79.47	103.50
rc105C5_0.75	233.90	233.90	388.23	345.83	127.05	127.05	135.00
rc108C5_0.75	305.52	**245.87**	439.57	**295.87**	69.30	**143.68**	146.25
rc204C5_0.75	176.00	176.00	915.24	920.88	65.06	65.06	99.75
rc208C5_0.75	174.38	174.38	292.39	300.10	0.00	0.00	0.75

Table 49: Non-linear charging comparison for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.75$

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c101C5_0.50	234.82	234.72	1623.73	1480.66	67.24	67.24	90.50
c103C5_0.50	165.67	165.67	1739.31	1764.25	57.02	57.02	62.50
c206C5_0.50	238.67	238.67	2420.98	2458.66	80.00	78.00	84.50
r104C5_0.50	137.01	137.01	334.02	339.04	44.91	44.91	54.50
r105C5_0.50	182.01	182.01	297.25	297.25	47.84	47.84	58.50
r202C5_0.50	143.04	143.04	228.91	235.88	53.77	53.77	54.50
r203C5_0.50	199.54	199.54	347.31	367.01	0.00	0.00	69.00
rc105C5_0.50	237.76	**235.67**	341.42	**381.84**	77.98	**60.81**	90.00
rc108C5_0.50	315.81	**245.87**	452.16	**299.59**	87.24	**96.39**	97.50
rc204C5_0.50	176.00	176.00	915.24	920.88	65.06	65.06	66.50
rc208C5_0.50	174.38	174.38	292.39	292.39	0.00	0.00	0.50

Table 50: Non-linear charging comparison for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.50$

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c101C5_0.25	278.47	**234.82**	1556.62	**1623.73**	33.17	**42.24**	45.25
c103C5_0.25	175.37	175.37	1678.44	1703.37	19.72	19.72	31.25
c206C5_0.25	242.56	242.56	1823.14	1971.17	24.35	24.35	42.25
c208C5_0.25	174.82	174.82	2029.29	2029.29	0.00	0.00	0.25
r104C5_0.25	137.01	137.01	334.02	339.04	0.00	0.00	27.25
r105C5_0.25	195.31	195.31	347.52	357.85	0.00	0.00	29.25
r202C5_0.25	143.13	143.13	415.98	415.98	0.00	0.00	27.25
r203C5_0.25	199.54	199.54	347.31	367.01	0.00	0.00	34.50
rc105C5_0.25	239.46	239.46	382.85	400.52	0.00	0.00	45.00
rc204C5_0.25	185.16	185.16	307.37	322.25	0.00	0.00	33.25
rc208C5_0.25	174.38	174.38	292.39	300.1	0.00	0.00	0.25

Table 51: Non-linear charging comparison for instances with $|\mathcal{N}| = 5$ and $\alpha = 0.25$

I Comparison results for non-linear charging for $|\mathcal{N}| = 10$

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c104C10_0.75	317.55	**271.37**	2207.55	**1653.1**	24.83	**122.53**	153.75
c202C10_0.75	282.38	282.38	5399.01	5400.93	72.62	72.62	138.75
r102C10_0.75	314.67	**289.81**	581.29	**654.87**	106.24	**90.82**	143.25
r103C10_0.75	193.90	**190.86**	443.38	**476.81**	100.85	**103**	113.25
r201C10_0.75	245.89	**231.30**	890.47	**1092.38**	90.38	**115.85**	118.5
r203C10_0.75	222.64	222.64	704.68	704.68	0.00	0.00	126.0
rc108C10_0.75	346.25	346.25	561.95	573.83	172.10	117.48	195.75
rc201C10_0.75	416.77	416.77	1164.19	1115.51	95.90	130.59	181.50
rc205C10_0.75	349.16	349.16	1093.69	1130.26	116.54	33.07	190.50

Table 52: Non-linear charging comparison for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.75$

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
c202C10_0.50	282.38	282.38	5399.01	5399.01	72.62	72.62	92.50
r102C10_0.50	319.06	**289.81**	580.20	**654.87**	49.99	**90.82**	95.50
r103C10_0.50	199.31	**198.77**	475.74	**405.94**	48.91	**51.18**	75.50
r201C10_0.50	235.13	**221.83**	1044.38	**1104.81**	73.25	**66.91**	79.00
r203C10_0.50	233.21	**222.64**	757.80	**704.68**	0.00	**0.00**	84.00
rc201C10_0.50	360.35	**358.03**	1204.13	**1544.69**	116.38	**52.14**	121.0
rc205C10_0.50	351.20	351.20	1132.33	1132.33	0.00	0.00	127.00

Table 53: Non-linear charging comparison for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.50$

Instance	Dist (lin)	Dist (non-lin)	Time (lin)	Time (non-lin)	Pol (lin)	Pol (non-lin)	Pol (max)
r103C10_0.25	216.52	**206.51**	448.10	**445.29**	18.26	**0.00**	37.75
r201C10_0.25	278.55	**204.94**	1010.72	**1049.32**	0.00	**37.8**	39.5
r203C10_0.25	222.64	222.64	704.68	704.68	0.00	0.00	42.00
rc201C10_0.25	362.01	**359.11**	1548.21	**1425.82**	0.00	**49.32**	60.5
rc205C10_0.25	349.18	349.18	1105.80	1093.69	0.00	23.91	63.50

Table 54: Non-linear charging comparison for instances with $|\mathcal{N}| = 10$ and $\alpha = 0.25$

J Code explanation

In general, in both `IteratedLocalSearch` and `IteratedLocalSearchNonLinearCharge`, a solution consists of an ArrayList of ArrayLists with Integers where a number of conventional routes follow first, where 0 is the depot and 1, 2, 3, ... is the first, second, third, etc. customer from the list in the instance. This is followed by 'null' to make a distinction between conventional and electric routes. Following are a number of electrical routes. Here a charging station is denoted by thousands, so the first charging station is denoted by 1000, the second by 2000, etc. Furthermore, when checking if the route is feasible in energy or when allocating charging stations, we assume full charging. When we check if the time is feasible, we calculate how much there is partially recharged by calculating the energy needed to move from a charging station to the next charging station or the end depot.

In `IteratedLocalSearch`, the time and pollution values are respectively linear and piecewise constant. The methods with 'realistic' are called to get the values for the non-linear and linear cases. However, these are not used at all in generating a solution.

In `IteratedLocalSearchNonLinearCharge`, we do generate the solution by calling the 'realistic' methods. For instance, when we check for feasibility, we check `isFeasibleRealisticTime` instead of `isFeasibleTime`. The pollution function remains piecewise constant.

We applied useful comments to understand the code to `IteratedLocalSearch`. Furthermore, in `IteratedLocalSearchNonLinearCharge`, we applied comments that denote the difference between the two classes. All these comments start with 'difference:', such that they are easy to find.