An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem

Wen, M.; Linde, Esben; Røpke, Stefan; Mirchandani, P.; Larsen, Allan

Published in:
Computers and Operations Research

Link to article, DOI:
10.1016/j.cor.2016.06.013

Publication date:
2016

Document Version
Peer reviewed version

An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem

M. Wen[a], E. Linde[b], S. Ropke[c], P. Mirchandani[d], A. Larsen[b]

[a]*Department of Mathematical Science, Xi'an Jiaotong-Liverpool University, 111 Ren Ai Road, Suzhou, Jiangsu, 215123, China*
[b]*Department of Transport, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[c]*Department of Management Engeering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[d]*School of Computing, Informatics, and Decision Systems Engineering, Arizona State University*

Abstract

This paper addresses the Electric Vehicle Scheduling Problem (E-VSP), in which a set of timetabled bus trips, each starting from and ending at specific locations and at specific times, should be carried out by a set of electric buses or vehicles based at a number of depots with limited driving ranges. The electric vehicles are allowed to be recharged fully or partially at any of the given recharging stations. The objective is to firstly minimize the number of vehicles needed to cover all the timetabled trips, and secondly to minimize the total traveling distance, which is equivalent to minimizing the total deadheading distance. A mixed integer programming formulation as well as an Adaptive Large Neighborhood Search (ALNS) heuristic for the E-VSP are presented. ALNS is tested on newly generated E-VSP benchmark instances. Result shows that the proposed heuristic can provide good solutions to large E-VSP instances and optimal or near-optimal solutions to small E-VSP instances.

Keywords: electric vehicles, vehicle scheduling, partial charging, large neighborhood search

1. Introduction

In recent years, a growing public concern about greenhouse gas emissions and health related pollution from the transportation sector has led to more attention to electric and other alternative fueled vehicles both in academies and industry (World Health Organization). Large vehicles such as buses contribute largely to this issue. For instance, in Copenhagen, Denmark, buses travel approximately 110 million kilometers a year and the bus fleet on average produces around 0.9 kg CO2 per kilometer along with other pollutants (Movia). Moreover, buses operate mainly in urban areas with dense population and therefore cause the greatest impact on health. If the public transit switched to zero-emission electric buses, the pollution in the city could be reduced significantly.

However, it is not trivial to substitute conventional buses with electric buses due to existing disadvantages of battery driven vehicles, e.g., limited battery capacity and long recharging time. These limitations result in 'range anxiety', which is the fear of running out of battery and the concern of making an unplanned trip (Bakker (2011)).

On the other hand, within commercial transport, a high degree of planning could be expected especially in the schedule based transportation sector, which carries out schedules with high punctuality according to a timetable. This suggests that electric vehicles have a good potential to be used for urban bus operations. However, to better utilize electric buses, the

Email address: min.wen@xjtlu.edu.cn (M. Wen)

above mentioned limitations must be taken into consideration during planning. This is the motivation of studying the Electric Vehicle Scheduling Problem (E-VSP) in this work.

The E-VSP, in which a set of timetabled trips should be assigned to a set of electric vehicles with limited driving ranges based at different depots, is an extension of the well-known Vehicle Scheduling Problem (VSP). The E-VSP can be described as a Multi-Depot VSP with distance constraints and charging possibilities. In the E-VSP, each trip starts from and ends at specific locations at predefined times. Each vehicle can be recharged fully or partially at any given recharging station. The recharging time is assumed to be a linear function of the amount of charged battery.

An E-VSP solution is a set of vehicle schedules, where each vehicle starts from and ends at its base depot, each trip is covered by exactly one vehicle and the vehicles' driving ranges are not exceeded. The objective is to first minimize the number of vehicles used and secondly minimize the total distance traveled. As the traveling distance of each trip is fixed, minimizing the total traveling distance is equivalent to minimizing the distance between the depot and the trip and between any two trips in the schedule, also known as deadheading distance.

The VSP has been extensively studied in the literature and extended to different variants, including the Multi-Depot VSP (MD-VSP) (Bodin et al. (1983) and Carpaneto et al. (1989)), the Multiple Vehicle Types VSP (Lenstra and Kan (1981)), and the VSP with Route Constraints (VSP-RC) (Bunte and Kliewer (2009)) where different types of route constraints can be enforced, including route duration (Freling and Paixao (1995)), route distance (Bodin et al. (1983)) or maximum vehicle bus line changes (Kliewer et al. (2008)). All the above mentioned VSP variants consider conventional vehicles and none of them allows recharging. A variant of the VSP that considers recharging/refueling options is the Alternative Fuel Vehicle Scheduling Problem (AF-VSP) studied by Adler (2014). In his problem, the alternative fuel vehicles are allowed to be refueled at given recharging stations to prolong the total distance the vehicles can travel. However, the AF-VSP is different from our E-VSP in the following aspects: 1) The AF-VSP only considers full charging. The vehicle's fuel level is set to full after visiting any recharging station; whereas the E-VSP considers partial charging, and introduces an extra decision on the necessary charging amount for each visit at any recharging station; 2) the charging time in the AF-VSP is fixed regardless of the remaining fuel level; whereas our charging time is assumed to be a linear function of the charged amount. In other words, the AF-VSP is a special case of our E-VSP. In Adler (2014), the author propose a construction heuristic as well as a column generation approach to solve the AF-VSP, and tests his algorithms on the metropolitan bus system of Phoenix, Arizona.

Another thread of literature relevant to the E-VSP is the Vehicle Routing Problem (VRP) for electric/alternative fuel vehicles, where the vehicles are used to serve a set of customers instead of the timetabled trips. Erdoğan and Miller-Hooks (2012) introduce the Green Vehicle Routing (G-VRP). Similar to Adler (2014), they also assume full charging and a fixed charging time for each visit to the recharging station. Felipe et al. (2014) consider an extension of the G-VRP, where recharging stations are of different types with different costs and recharging speeds. Moreover, partial charging is allowed and the recharging time is assumed to be a linear function of the amount of energy recharged. Schneider et al. (2014) extend the G-VRP to the Electric Vehicle Routing Problem with Time Windows (E-VRPTW) by considering customers' time windows, unlimited number of recharges per route and a variable recharging time which depends on the remaining fuel level when a vehicle arrives at the recharging station. However, partial charging is still not an option in Schneider et al. (2014), i.e., a vehicle must be fully recharged when leaving the recharging station. Hiermann et al. (2016) study the Electric Fleet Size and Mix VRPTW, where electric vehicles with different battery capacities, load capacities,

energy consumptions and recharging rates are used. In Goeke and Schneider (2015) a mixed fleet of electric vehicles and conventional vehicles is used. Desaulniers et al. (2014) further extend the E-VRPTW by allowing partial charging. They develop branch-price-and-cut-algorithms to solve different variants of the problem with single/multiple-recharge and full/partial-rechange. They test their algorithms on benchmark instances and demonstrate the benefit of multiple-recharge and partial-recharge.

In this work, we consider the electric scheduling problem with partial recharging described in Section 2, and present a mathematical model (Section 3) as well as an Adaptive Large Neighborhood heuristic (Section 4) for solving this problem. The heuristic is tested on newly generated E-VSP instances. Computational results are provided in Section 5, followed by a conclusion and future work in Section 6.

2. Problem description

The input to the E-VSP includes a set of timetabled trips, a set of vehicles, a set of depots and a set of recharging stations. Each trip has a specific start time, end time, start location, end location and traveling distance. Each vehicle has a limited driving range, and should start from and end at its base depot. The vehicle's fuel consumption is assumed to be a linear function of the traveling distance. The consumption rate. i.e, the amount of fuel consumed per unit distance, is given. The vehicles can visit any recharging station to recharge any amount up to the full battery capacity. The charging time is assumed to be a linear function of the battery charge gained and the charging rate, i.e., the time needed for charging one unit battery, is also given. The distance and the time for the deadheading traveling between any two timetabled trips and between any depot/station and any timetabled trip are known. The problem is to find a set of least-cost schedules for the vehicles to perform the timetabled trips. Each trip should be covered by exactly one schedule that is performed by one vehicle. The charge level of the vehicle must be non-negative at any time throughout the schedule. The cost consists of a relatively high pullout cost of using each vehicle and an operational traveling cost.

To illustrate this problem, a small example consisting of two timetabled trips, one recharging station and one depot is given in Fig. 1. The start time, end time, start location and end location of the trips are given in Tab. 1. The travel time and the distance of each arc are assumed to be the same, and are given in the figure. The vehicle range is 30. Both the consumption rate and the charging rate are 1. The pullout cost is 1000 and the traveling cost is 1 per unit distance. The optimal solution to this example is to use a single vehicle to perform the schedule given in Tab. 2. After performing trip 1, the vehicle visits the recharging station with a remaining battery of 5. According to the timetable, there is a surplus of time to recharge partially for extra 20 units, which prolongs its driving distance by 10 units and makes it possible for the vehicle to carry out the rest of the schedule without running out of battery.

As can be noticed, for the same example, if recharging is not allowed or if only full charging is allowed, two vehicles will be needed and the total cost will be 2060. This illustrates the benefit of allowing partial charging in the scheduling problem for electric vehicles.

Figure 1: A small E-VSP example with single depot, two trips and one recharging station.

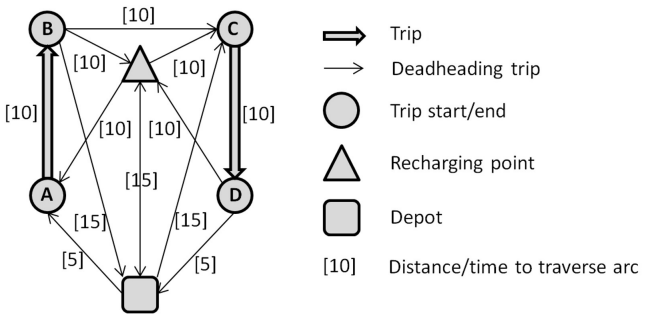


Table 1: The timetables associated with the two trips in the small example.

Trip	Origin	Time	Destination	Time
1	A	07:50	B	08:00
2	C	08:40	D	08:50

3. Mathematical formulation

A mathematical model for the E-VSP can be derived from the model for the E-VRPTW presented in Schneider et al. (2014). In fact, if one does not need to model partial recharging and multiple depots then the E-VRPTW model by Schneider et al. (2014) is sufficient to model the E-VSP. Adding partial recharging and multiple depots to the model is no big feat. However, the resulting model is very difficult to solve just like the original E-VRPTW model. We implemented such a model and even instances with just 10 trips proved to be challenging.

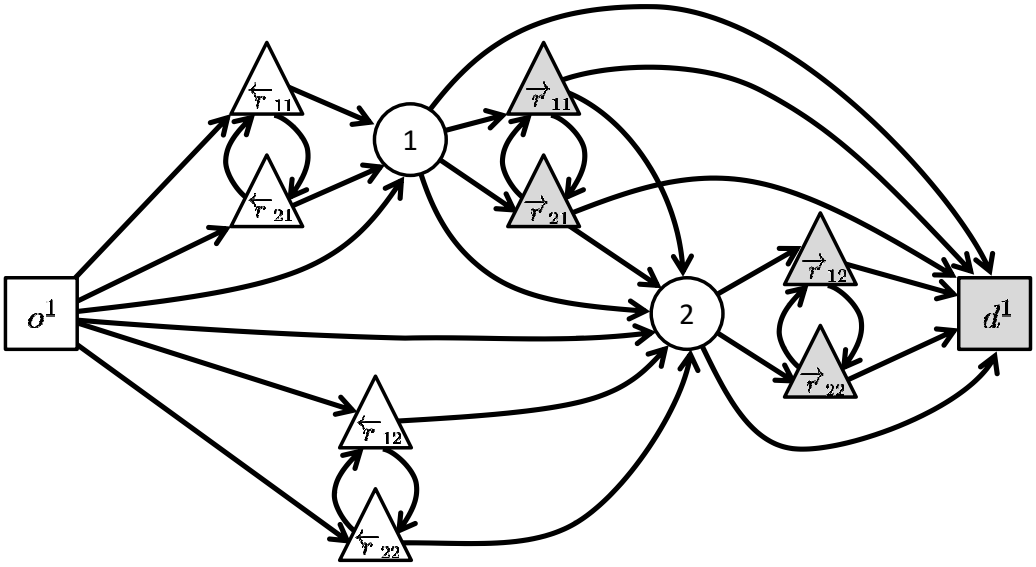
In this section we present an improved model that takes advantage of the fact that the start time of each trip in the E-VSP is fixed. By making copies of the recharging stations and assigning each copy to a specific trip it is possible to stipulate specific time windows on each recharging station copy that ensure that the underlying graph structure is close to a directed acyclic graph (DAG). Cycles only occur between nodes representing copies of recharging stations assigned to the same customer. The DAG-like structure ensures that the solution to the LP relaxation of the model does not contain paths from a later trip to an earlier trip. In the following, we make this clearer and show an example of the underlying graph for a simple example.

The model is defined on a directed graph $G = (V,A)$, where V is the set of nodes and A is the set of feasible arcs. Each timetabled bus trip is represented by a node in the graph. Let $T \subseteq V$ denote the set of trip nodes, each of which is associated with a fixed start time a_i, a service time s_i and a fixed battery consumption e_i. Let D be the set of depots. Each depot $\beta \in D$ is represented by two nodes in the graph: o^β (start node for depot β) and d^β (end node for depot β). Let $\bar{D} \in V$ be the set of all start and end depot nodes. Let R denote the set of recharging stations. The recharging stations are being represented by a set of duplicate nodes $\bar{R}$ in the graph G, with cardinality $\bar{R} = 2|R||T|$. For each combination of a recharging station $i \in R$ and trip $j \in T$ two nodes are being created. One node denoted $\overrightarrow{r}_{ij}$ can be visited just after trip j and one node $\overleftarrow{r}_{ij}$ can only be visited on the path from a depot to j if j is the first trip on its route. The only nodes that can be visited between node j and $\overrightarrow{r}_{ij}$ are nodes from $\{\overrightarrow{r}_{kj} \in \bar{R} : k \in R \setminus \{i\}\}$. Each node in $\bar{R}$ can be visited at most once. The cost, distance and the traveling time of each arc $(i, j) \in A$ are given by c_{ij}, d_{ij} and t_{ij}, respectively. In the model we will often use the notation t_a for an arc $a = (i, j)$ instead of t_{ij} and similar notation for c and d. It is assumed that $t_{ik} \leq t_{ij} + t_{jk}$ for all $i \in T, j,k \in \bar{R}$ and for all $i, j \in \bar{R}, k \in T$, and likewise for d_{ij} and c_{ij}. This assumption is denoted

Table 2: The optimal solution to the small example.

Location	Arrival time	Departure time	Arrival fuel level	Departure fuel level	Accumulated cost
Depot	-	7:45	-	30	0
A	7:50	7:50	25	25	5
B	8:00	8:00	15	15	15
Recharging station	8:10	8:30	5	25	25
C	8:40	8:40	15	15	35
D	8:50	8:50	5	5	45
Depot	8:55	-	0	0	50

Figure 2: An illustration of the graph G used in the mathematical model for an instance with 1 depot, 2 recharging stations and 2 trips.



the *partial triangle inequality assumption*. One cannot assume that a triangle inequality holds for the travel times (or cost or distances) for all nodes in V since each trip node models driving a certain distance. Each depot and station $i \in D \cup R$ has a time window $[a_i, b_i]$, within which the vehicle can visit. From the station time windows, we can derive tighter time windows for the nodes in R due to the partial triangle inequality assumption and since we know that each node in $\bar{R}$ is visited before/after a certain trip. The set of arcs A is a subset of $V \times V$. Many arcs can be eliminated from A because of the fixed starting times for nodes in T and because of the rules for when each node in $\bar{R}$ can be visited (outlined above). Figure 2 shows an example of the graph structure for an instance with two trips, two recharging points and a single depot. The circles represents the two trips, the squares represent the start and end depot. Triangles represent the nodes in $\bar{R}$ with white nodes represent nodes $\overleftarrow{r}_{ij}$ and grey nodes representing $\overrightarrow{r}_{ij}$. In the figure, it is assumed that the starting time of trip 2 is later than that of trip 1.

Let $\Delta^+(i)$ and $\Delta^-(i)$ denote the set of arcs that can originates in node i and that ends in node i, respectively. We assume that the vehicle fleet is homogeneous. Let Q denote the driving range, r the fuel consumption rate (the fuel units spent per unit distance) and g the recharging rate (the time for charging one unit battery). Let p be the pullout cost associated with a vehicle.

Let binary decision variable x_a^β equal 1 if a vehicle from depot $\beta \in D$ traverses arc $a \in A$ and let non-negative variable

h_i be the amount of energy that the vehicle charges during the visit to station copy $i \in \bar{R}$. The charged amount h_i depends on the remaining battery charge and the length of time that vehicle stays at station i. Hence two extra non-negative variables, z_i and y_i, are defined for tracking the time that a vehicle starts the service/charging at node i and the remaining battery charge of the vehicle when it arrives at node i. The model uses constants $M_{ij}^1, M_{ij}^2, M_{ij}^3$ and M_{ij}^4 in big-M constraints, the values for these constants will be explained toward the end of this section. The mathematical model can be formulated as follows:

$$\text{minimize} \sum_{\beta \in D} \sum_{a \in A} c_a x_a^\beta + p \sum_{\beta \in D} \sum_{a \in \Delta^+(o^k)} x_a^\beta \tag{3.1}$$

subject to

$$\sum_{\beta \in D} \sum_{a \in \Delta^+(i)} x_a^\beta = 1 \qquad \forall i \in T \tag{3.2}$$

$$\sum_{a \in \Delta^+(i)} x_a^\beta \leq 1 \qquad \forall i \in \bar{R},\ \beta \in D \tag{3.3}$$

$$\sum_{\beta' \in D \setminus \{\beta\}} \sum_{a \in \Delta^+(o^\beta)} x_a^{\beta'} = 0 \qquad \forall \beta \in D \tag{3.4}$$

$$\sum_{\beta' \in D \setminus \{\beta\}} \sum_{a \in \Delta^-(d^\beta)} x_a^{\beta'} = 0 \qquad \forall \beta \in D \tag{3.5}$$

$$\sum_{a \in \Delta^+(i)} x_a^\beta - \sum_{a \in \Delta^-(i)} x_a^\beta = 0 \qquad \forall i \in T \cup \bar{R},\ \beta \in D \tag{3.6}$$

$$z_i = a_i \qquad \forall i \in T \tag{3.7}$$

$$a_i \leq z_i \leq b_i \qquad \forall i \in \bar{D} \cup \bar{R} \tag{3.8}$$

$$z_i + s_i + t_a - M_{ij}^1 (1 - \sum_{\beta \in D} x_a^\beta) \leq z_j \qquad \forall i \in T, a = (i,j) \in \Delta^+(i) \tag{3.9}$$

$$z_i + t_a + g \cdot h_i - M_{ij}^2 (1 - \sum_{\beta \in D} x_a^\beta) \leq z_j \qquad \forall i \in \bar{R}, a = (i,j) \in \Delta^+(i) \tag{3.10}$$

$$y_i - e_i - r \cdot d_a + M_{ij}^3 (1 - \sum_{\beta \in D} x_a^\beta) \geq y_j \qquad \forall i \in T \cup \{o^\beta : \beta \in D\},\ a = (i,j) \in \Delta^+(i) \tag{3.11}$$

$$y_i + h_i - r \cdot d_a + M_{ij}^4 (1 - \sum_{\beta \in D} x_a^\beta) \geq y_j \qquad \forall i \in \bar{R},\ a = (i,j) \in \Delta^+(i) \tag{3.12}$$

$$0 \leq h_i \leq Q - y_i \qquad \forall i \in \bar{R} \tag{3.13}$$

$$0 \leq y_i \leq Q \qquad \forall i \in V \tag{3.14}$$

$$x_a^\beta \in \{0,1\} \qquad \forall a \in A,\ \beta \in D \tag{3.15}$$

The objective is to minimize the total cost, including the traveling cost and vehicle cost. Constraints (3.2) enforce each trip to be covered exactly once. Constraints (3.3) make sure that each station node in $\bar{R}$ is visited at most once by each vehicle. Constraints (3.4) and (3.5) ensure the compatibility between variable and depots and constraints (3.6) are flow conservation constraints. Constraints (3.7) and (3.8) are the time window constraints, enforcing that the depot and station nodes are visited within their service time window and the trip nodes are performed at their specific start times. Constraints (3.9) and (3.10) keep track of the visit time at each node. If a vehicle travels from i to j, then $z_i + s_i + t_{ij} \leq z_j$

for $i \in T$ and $z_i + t_{ij} + g \cdot h_i \leq z_j$ for $i \in R$, where $g \cdot h_i$ is the time needed to recharge h_i at station i. Similarly, constraints (3.11) and (3.12) keep track of the remaining battery of each vehicle when it arrives at any node. Constraints (3.13) impose that the battery cannot be recharged beyond its capacity. Constraints (3.14) are the battery capacity constraints and constraints (3.15) define the binary variables.

For most of the other VSP variants, such as MD-VSP and VSP-RC, the visit time of each vehicle at each node is not modeled in their formulations because the infeasible arcs (i, j), where $a_j \leq a_i + s_i + t_{ij}$, are already removed from the graph. Hence, the solution will always be feasible with respect to time. However, in our problem, since we introduce the partial recharging, one has to model the visit time in order to determine how long a vehicle can stay at a station and how much it can recharge. Therefore, the model for the E-VSP is more complicated than the other VSP variants. The MD-VSP is a special case of the E-VSP, where there is no limit on the driving range ($Q = \infty$) and hence provides a lower bound to the E-VSP.

There are several ways strengthening the inequalities (3.9)-(3.12) based on the methods proposed by Desrochers and Laporte (1991). We have attempted the following inequalities:

$$z_j \geq a_j + \sum_{a=(i,j)\in A} \left(\max\{0, a_i - a_j + t_a + s_i\} \sum_{\beta \in D} x_a^\beta \right) \quad \forall j \in \bar{R} \tag{3.16}$$

and

$$z_i \leq b_i - \sum_{a=(i,j)\in A} \left(\max\{0, b_i - b_j + t_a + s_i\} \sum_{\beta \in D} x_a^\beta \right) \quad \forall i \in \bar{R} \tag{3.17}$$

which are straightforward adaptations of inequalities (28) and (29) in Desrochers and Laporte (1991).

It is possible to enforce tighter limits on the remaining battery charge y_i compared to the limits given by (3.14). In general we can compute a lower γ_i and upper bound δ_i on the y_i for each node $i \in \bar{R} \cup T$ and exchange (3.14) by

$$\gamma_i \leq y_i \leq \delta_i$$

The lower bound γ_i is based on a minimum driving that needs to be carried out after visiting node i before reaching a recharging station or end depot while the upper bound δ_i is based on the minimum driving from the last recharging station/depot before reaching node i. With these bounds a valid inequalities involving y_i can again be derived from (28) and (29) in Desrochers and Laporte (1991):

$$y_i \geq \gamma_i + \sum_{a=(i,j)\in A} \left(\max\{0, \gamma_j - \gamma_i + r d_a + e_i\} \sum_{\beta \in D} x_a^\beta \right) \quad \forall i \in T$$

$$\begin{aligned} y_j \leq \delta_j - & \sum_{\substack{a=(i,j)\in A \\ i \in T}} \left(\max\{0, \delta_j - \delta_i + r d_a + e_i\} \sum_{\beta \in D} x_a^\beta \right) + \\ & \sum_{\substack{a=(i,j)\in A \\ i \in \bar{R} \cup \{o^1, \ldots, o^{|D|}\}}} \left(\max\{0, \delta_j - \delta_i + r d_a\} \sum_{\beta \in D} x_a^\beta \right) \quad \forall j \in \bar{R} \cup T \end{aligned}$$

What is left to describe are the values for $M_{ij}^1, M_{ij}^2, M_{ij}^3$ and M_{ij}^4 in constraints (3.9) -(3.12): we use $M_{ij}^1 = b_i + s_i + t_{ij} - a_j$, $M_{ij}^2 = b_i + t_{ij} + gQ - a_j$, $M_{ij}^3 = \delta_j - (\gamma_i - e_i - r \cdot d_{ij})$ and $M_{ij}^4 = \delta_j - (\gamma_i - r \cdot d_{ij})$.

4. Adaptive Large Neighborhood Search

As the E-VSP is an NP hard problem and the goal of this work is to solve large instances, we developed an Adaptive Large Neighborhood Search (ALNS) heuristic to solve the E-VSP. ALNS, firstly proposed by Ropke and Pisinger (2006), has been successfully applied to VRPTW and various VRP extensions. The trips in our E-VSP are similar to the customers in a VRPTW. A trip with a fixed start time is treated as a customer with a very tight time window, where the earliest start time coincides with the latest start time. Since the start time is fixed, the graph of an E-VSP can be reduced significantly compared to a VRPTW with wide TWs (Haghani and Banihashemi (2002)). In the E-VSP, there are extra inputs, including the traveling distance and fuel consumption associated with each trip , and the extra set of stations that can be visited when necessary.

ALNS is an extension of Large Neighborhood Search (LNS) originally proposed by Shaw (1998). The basic idea of LNS is to search in large neighborhoods, which may contain more and potentially better solutions compared to small neighborhoods. The neighborhood of a solution is defined implicitly by a destroy method and a repair method. A destroy method disrupts part of the current solution while a repair method rebuilds the destroyed solution. In ALNS, a series of destroy and repair methods are employed. In each iteration, the destroy/repair methods to be applied on the current solution are selected according to their weights, which are adaptively adjusted based on the performance of the destroy/repair methods in the previous iterations. The destroy/repair methods that have performed well in the previous iterations are more likely to be chosen for the current iteration. The resultant solution is accepted based on a user defined acceptance criterion, and the heuristic stops when the stop criterion is met.

In our implementation, ALNS is run for multiple restarts. In each restart ALNS stops when a predefined maximum iteration, N_{max}, is reached. In each iteration, if the new solution is better than the current one, it is always accepted; otherwise, it is accepted with a probability $e^{((f(x)-f(x^{'}))/T)}$, where the temperature T is initialized by T_0 and decreased over time to T_1. This makes it less likely to accept poor solutions towards the end of the search. Additionally, we have introduced a diversification phase applied occasionally to further diversify the search. At the end of the heuristic, a post optimization is used to further improve the solution. The entire framework of our ALNS is described in Alg. 1. The constructive heuristic for generating the initial solution, the destroy/repair methods, the diversification phase and the post optimization phase are elaborated in Sections 4.1–4.4.

4.1. Initialization and diversification

The initial solution of for ALNS is generated by a greedy constructive heuristic, similar to the Concurrent Scheduler Algorithm for the AF-VSP proposed by Adler (2014) and originally designed by Bodin et al. (1978). The trips are first sorted in a non-decreasing order of their start times and iteratively added to the solution. In each iteration, we try to insert the selected trip to the end of an existing schedule that leads to the minimum increase in the objective value. If it is not feasible to be added to any existing schedule, a new schedule will be created for the trip. The procedure stops until all the trips are served.

A diversification phase, which modifies a larger part of a solution, is applied occasionally in the ALNS framework to diversify the search into different parts of the solution space. A solution is modified by firstly removing a large number of

Algorithm 1 Adaptive Large Neighborhood Search.

Input: A feasible solution x
$x^* \leftarrow x$
$T \leftarrow T_0$
repeat
  if apply diversification **then**
    $x \leftarrow$diversify(x)
  choose a destroy method and a repair method based on their weights
  $x' \leftarrow$ the resulting solution after applying the selected destroy and repair methods
  if accept(x', x) **then**
    $x \leftarrow x'$
    if $f(x') < f(x^*)$ **then**
      $x^* \leftarrow x$
  Adjust the weights of the destroy/repair mehtods if necessary
  $T \leftarrow \alpha T$
until Iteration N_{max} is reached
$x^* \leftarrow$ post-optimization() **return** x^*

random trips from the existing solution and later iteratively inserting a random trip back one by one to the solution at the best position.

4.2. Destroy Methods

A selection of destroy methods are implemented in our ALNS, including a random destroy, a time- related removal and a neighboring schedule based removal. The random destroy method simply removes a number of trips randomly from the solution. The time-related removal removes the trips having similar start/end times. It first removes a random trip, and then repeatedly removes the trip that has the closest start/end time to any of the selected trips. The procedure stops until a given number of trips are removed. The neighboring schedule based removal aims at removing similar and closely located schedules. A schedule s' is a neighboring schedule of another schedule s, if $s' = argmin_{\tilde{s} \in \bar{\Omega}} c(s, \tilde{s})$, where $c(s, \tilde{s}) = (\sum_{j \in \tilde{s}} \sum_{i \in s} d_{ij}) / (|s| \cdot |\tilde{s}|)$ is the average distance between the trips in s and in $\tilde{s}$, and $\bar{\Omega}$ is the set of the schedules excluding s in the current solution. The removal method removes a random schedule and its neighboring schedule.

At the end of all the abovementioned removal methods, single-trip schedules are additionally removed from the solution.

4.3. Repair Methods

When a solution is destroyed, a repair method is applied to rebuild it. We have implemented a class of regret insertion heuristics (Potvin and Rousseau (1993)). These methods inserts one trip to the best position at a time. They differ in their ways of selecting the trip to be inserted.

Let N^r denote the set of trips to be inserted, and $\Delta f_{i,p}$ the cost of inserting trip $i \in N^r$ to position p in the current partial solution. If inserting i to position p is feasible with respect to the driving range constraints (referred to as fuel feasible) and timetable constraints (referred to as time feasible), $\Delta f_{i,p} = d_{pre(p),i} + d_{i,suc(p)} - d_{pre(p),suc(p)}$, where $pre(p)$ and $suc(p)$ are the predecessor trip and successor trip corresponding to position p respectively. If the insertion leads to an infeasible solution with respect to the timetable constraints, we set $\Delta f_{i,p} = \infty$. If the insertion is time feasible but not fuel feasible, we try to insert a station before or after trip i to resolve the fuel infeasibility. If such a station exists, the best station s^* that leads to the minimum increase in the objective function is selected, and $\Delta f_{i,p}$ is set to $d_{pre(p),s^*} + d_{s^*,i} + d_{i,suc(p)} - d_{pre(p),suc(p)}$ if s^* is inserted before trip i and to $d_{pre(p),i} + d_{i,s^*} + d_{s^*,suc(p)} - d_{pre(p),suc(p)}$ if s^* is inserted after trip i.

Let c_i^1 denote the insertion cost of trip i at its best position, $c_i^1 = min_{p \in P} c_{ip}$, where P is the set of all the positions; and c_i^k the insertion cost at the kth best position. In a regret-k heuristic, the trip $i^* = argmax_{i \in N^r}(\sum_{j=1}^{j=k}(c_i^j - c_i^1))$ is selected as the next trip to be inserted.

Our ALNS uses regret-2, regret-3 and regret-4. All the heuristics are implemented in both deterministic version and stochastic version. In the deterministic version, the best trip is selected; whereas, in the stochastic version, a random trip from the l best trips is selected. We set l to $|N^r| \cdot R^\beta$, where R is a random number between $[0,1]$, and $\beta > 1$ is a user defined parameter.

4.4. Post optimization

Finally, a post optimization phase is implemented. The schedules in all the feasible solutions found by ALNS are stored and in the end given as the input to the following set partitioning model:

$$\text{minimize} \sum_{s \in \Omega} c_s x_s \tag{4.1}$$

subject to:

$$\sum_{s \in \Omega} a_s^i x_s = 1 \qquad \forall i \in T \tag{4.2}$$

$$\sum_{s \in \Omega} b_s^d x_s \leq n^d \qquad \forall d \in D \tag{4.3}$$

$$x_s \in \{0,1\} \qquad \forall s \in \Omega \tag{4.4}$$

where Ω denotes the set of input schedules found by ALNS, c_s is the cost of schedule $s \in \Omega$, a_s^i is a binary variable indicating if trip $i \in T$ is covered by schedule s, b_s^d is a binary variable indicating if schedule s is associated with depot d, n^d is the maximum number of vehicles from depot d, and the binary variable x_s equals 1 if schedule s is selected. The objective (4.1) is to minimize the total cost of chosen schedules. Constraints (4.2) make sure that each trip is covered by exactly one schedule. Constraints (4.3) enforce that the number of vehicle used at a depot does not exceed the limit and (4.4) defines the binary decision variable. The set partitioning model is solved by CPLEX 12.6. The solution to this model is at least as good as the best solution found by ALNS.

5. Computational results

The ALNS heuristic is implemented in a single thread Java environment and run on a PC with Windows 7, Intel Core i7-3520M, 2.9GHz, 8 GB RAM. The mathematical model is solved by CPLEX 12.6.

5.1. Data

A set of E-VSP benchmark instances are generated in a way similar to the generation of the MD-VSP instances in Carpaneto et al. (1989) and inspired by Pepin et al. (2009). The depots and the two ending points of the trips are uniformly distributed in a $60km \times 60km$ Euclidean plane. The start time of a trip is a random integer uniformly distributed with a probability of 15% in [420, 480], 70% in [480, 1020], and 15% in [1020, 1080]. The end time of a trip is a random integer uniformly distributed in (the trip's start time+the trip's distance+5, the trip's start time+the trip's distance+40). A set of recharging stations are uniformly distributed in the plane, and one recharging station is placed at every depot. The vehicle driving range is set to 150 km, which seems reasonable compared to the spatial distribution of the trips. The charging rate is 0.8 minute/km, which means that a complete charging from zero to full takes two hours. The generated instances are of different sizes.

Tab. 3 describes the 14 classes of instances that were generated. For each class we list the number of trips, the number of depots, the number of stations as well as the number of instances generated. The small instances with up to 30 trips are solved by both ALNS and CPLEX, whereas the large instances only are solved by ALNS. All instances are available on-line as supplementary material.

Table 3: The E-VSP instances.

Name	# Depots	# station	# trips	# instances
D2_S2_C10	2	2	10	5
D2_S4_C10	2	4	10	5
D2_S2_C15	2	2	15	5
D2_S4_C15	2	4	15	5
D2_S2_C20	2	2	20	5
D2_S4_C20	2	4	20	5
D2_S2_C25	2	2	25	5
D2_S4_C25	2	4	25	5
D2_S2_C30	2	2	30	5
D2_S4_C30	2	4	30	5
D2_S4_C100	2	4	100	5
D4_S8_C100	4	8	100	5
D4_S8_C500	4	8	500	5
D8_S16_C500	8	16	500	5

5.2. ALNS parameter setting

A single run of the algorithm consists of five restarts of the ALNS heuristic. In each restart, ALNS executes 50,000 iterations. A simple parameter tuning process has been carried out using a small set of instances including D2_S4_C100_1, D4_S8_C100_1, D4_S8_C500_1, D4_S8_C500_2, D8_S16_C500_1 and D8_S16_C500_2. This leads to the following parameter configuration: The initial temperature at the first iteration is 50 and the end temperature at the last iteration is 0.5.

Therefore the cooling rate α is set to 0.99991. In the random removal and time–related removal, the number of trips to be removed from the current solution is chosen randomly between 1 and 20. When the repair method involves a stochastic element, the parameter β is chosen to be 2. The solution is diversified every 2400 iterations. In each diversification, 50% of all the trips are removed from the solution. The time limit given to the post optimization is 150 seconds in each ALNS restart. To show the effect of the destroy/repair methods, we have presented the relative changes in the objective value when one destroy/repair method is excluded from the algorithm in Table 4. Results show that the random trip removal and neighboring schedule based removal are the two methods contributed the most to the solution quality. Excluding a repairing method in general does not lead to a much different solution. However, we chose to keep all of them.

Table 4: Relative changes in the objective value when one destroy/repair method is excluded.

Excluded destroy/repairing method	Change in the objective value
time-related removal	+0.84%
random trip removal	+2.01%
random neighboring schedule removal	+3.16%
regret-2 insertion	+0.17%
regret-3 insertion	+0.07%
regret-4 insertion	+0.04%

5.3. Results on the small instances

Tables 5 and 6 present the E-VSP solutions given by the ALNS and CPLEX (using the model presented in Section 3), and the corresponding MD-VSP lower bound computed by CPLEX. The difference between the E-VSP and the MD-VSP is that the driving range is 150 km in the former and unlimited in the latter. The E-VSP model was solved on a Intel Core I7-2620M CPU running at 2.70 GHz and having two cores (4 virtual cores through hyper-threading). CPLEX version 12.6 was used and CPLEX was allowed to take advantage of parallel processing and a time limit of 1 hour was enforced.

For the E-VSP MIP model we report the solution obtained (UB), this solution is optimal as long as the time limit was not reached. The best lower bound (LB) obtained is reported when the instance is not solved to optimality, the number of vehicles used in the solution (v), the dead heading distance (d.h) and the time spent in seconds (t (s)). A dash in the time column indicate that the instance was not solved to optimality within the time limit.

For the ALNS heuristic and MDVSP lower bound we report similar information. The second to last column in the tables reports the percentage gap between best MIP solution (z^{MIP}) and best ALNS solution (z^{ALNS}), this number is calculated as

$$\frac{z^{ALNS} - z^{MIP}}{z^{MIP}} \cdot 100.$$

Similarly the last column compares the ALNS solution to the lower bound obtained by solving the MD-VSP (z^{MDVSP}). This number is calculated as follows

$$\frac{z^{ALNS} - z^{MDVSP}}{z^{MDVSP}} \cdot 100.$$

As can be seen from the table, ALNS is able to find high quality solutions to the small instances containing up to 30 trips. The computational time of ALNS is just a fraction of the time needed CPLEX for the more challenging instances. The comparison to the MD-VSP results shows that solving the MD-VSP often does not provide a tight lower bound to the E-VSP. This is caused by the restricted driving range in the E-VSP that forces vehicles to recharge which increases deadheading and decreases utilization.

	E-VSP MIP					E-VSP ALNS				MDVSP				MIP gap	LB gap
name	UB	LB	#v.	d.h.	t (s)	UB	#v.	d.h.	t (s)	Opt	#v.	d.h	t (s)	%	%
D2_S2_C10_0	40373.8	-	4	373.8	0.4	40373.8	4	373.8	4.2	30336.1	3	336.1	0.1	0.00	33.09
D2_S2_C10_1	40321.2	-	4	321.2	0.2	40321.2	4	321.2	3.8	40276.5	4	276.5	0.1	0.00	0.11
D2_S2_C10_2	50435.1	-	5	435.1	0.6	50435.1	5	435.1	3.6	40386.3	4	386.3	0.1	0.00	24.88
D2_S2_C10_3	30331.3	-	3	331.3	0.3	30331.3	3	331.3	3.8	30267.9	3	267.9	0.1	0.00	0.21
D2_S2_C10_4	30316.0	-	3	316	0.2	30316.0	3	316.0	3.7	30270.6	3	270.6	0.1	0.00	0.15
D2_S4_C10_0	30326.6	-	3	326.6	0.5	30326.6	3	326.6	3.8	30314.3	3	314.3	0.1	0.00	0.04
D2_S4_C10_1	40346.4	-	4	346.4	3.5	40346.4	4	346.4	3.4	30328.5	3	328.5	0.1	0.00	33.03
D2_S4_C10_2	30383.1	-	3	383.1	0.5	30383.1	3	383.1	3.8	30351.2	3	351.2	0.1	0.00	0.11
D2_S4_C10_3	40241.5	-	4	241.5	0.4	40241.5	4	241.5	3.0	40235.5	4	235.5	0.1	0.00	0.01
D2_S4_C10_4	40335.1	-	4	335.1	0.5	40335.1	4	335.1	3.2	40302.9	4	302.9	0.1	0.00	0.08
D2_S2_C15_0	40484.6	-	4	484.6	0.6	40484.6	4	484.6	5.7	40418.6	4	418.6	0.1	0.00	0.16
D2_S2_C15_1	50465.4	-	5	465.4	0.3	50465.4	5	465.4	5.4	50399.8	5	399.8	0.1	0.00	0.13
D2_S2_C15_2	50730.5	-	5	730.5	1.0	50730.5	5	730.5	5.7	50488.5	5	488.5	0.1	0.00	0.48
D2_S2_C15_3	50367.3	-	5	367.3	0.5	50367.3	5	367.3	5.1	50318.4	5	318.4	0.1	0.00	0.10
D2_S2_C15_4	50441.7	-	5	441.7	0.5	50441.7	5	441.7	5.2	50375.4	5	375.4	0.1	0.00	0.13
D2_S4_C15_0	50500.0	-	5	500	8.6	50500.0	5	500.0	4.6	50438.1	5	438.1	0.1	0.00	0.12
D2_S4_C15_1	60478.1	-	6	478.1	6.6	60478.1	6	478.1	5.2	50480.2	5	480.2	0.1	0.00	19.81
D2_S4_C15_2	40521.1	-	4	521.1	1.0	40521.1	4	521.1	5.6	40493.3	4	493.3	0.1	0.00	0.07
D2_S4_C15_3	60403.0	-	6	403	0.9	60403.4	6	403.4	5.1	60386.1	6	386.1	0.2	0.00	0.03
D2_S4_C15_4	50396.2	-	5	396.2	8.6	50396.2	5	396.2	4.7	50368.5	5	368.5	0.1	0.00	0.06
D2_S2_C20_0	60681.0	-	6	681	0.5	60681.0	6	681.0	6.1	60536.6	6	536.6	0.1	0.00	0.24
D2_S2_C20_1	60580.2	-	6	580.2	0.6	60580.2	6	580.2	7.1	60506.5	6	506.5	0.1	0.00	0.12
D2_S2_C20_2	60832.4	-	6	832.4	13.9	60832.4	6	832.4	7.7	50650.5	5	650.5	0.1	0.00	20.10
D2_S2_C20_3	70429.9	-	7	429.9	0.7	70429.9	7	429.9	7.4	70401.4	7	401.4	0.1	0.00	0.04
D2_S2_C20_4	70535.3	-	7	535.3	0.7	70535.3	7	535.3	7.3	70472.2	7	472.2	0.1	0.00	0.09
D2_S4_C20_0	50614.7	-	5	614.7	23.4	50614.7	5	614.7	7.3	50539.8	5	539.8	0.1	0.00	0.15
D2_S4_C20_1	60511.7	-	6	511.7	18.3	60511.7	6	511.7	7.0	50513.3	5	513.3	0.1	0.00	19.79
D2_S4_C20_2	50672.9	-	5	672.9	5.4	50674.3	5	674.3	7.5	50627.2	5	627.2	0.1	0.00	0.09
D2_S4_C20_3	60487.1	-	6	487.1	1.6	60487.1	6	487.1	6.7	60453.2	6	453.2	0.2	0.00	0.06
D2_S4_C20_4	70503.3	-	7	503.3	13.2	70503.3	7	503.3	8.1	70483	7	483.0	0.1	0.00	0.03

Table 5: Results on E-VSP instances containing 10 to 20 trips.

	E-VSP MIP					E-VSP ALNS				MDVSP				MIP gap	LB gap
name	UB	LB	#v.	d.h.	t (s)	UB	#v.	d.h.	t (s)	Opt	#v.	d.h	t (s)	%	%
D2_S2_C25_0	70833.6	-	7	833.6	1.3	70833.6	7	833.6	9.8	70639.5	7	639.5	0.2	0.00	0.27
D2_S2_C25_1	60695.6	-	6	695.6	11.9	60695.6	6	695.6	7.3	60537.1	6	537.1	0.1	0.00	0.26
D2_S2_C25_2	70934.7	-	7	934.7	40.7	70934.7	7	934.7	7.9	60754	6	754.0	0.1	0.00	16.76
D2_S2_C25_3	70621.5	-	7	621.5	13.6	70625.9	7	625.9	8.6	70516.7	7	516.7	0.2	0.01	0.15
D2_S2_C25_4	80623.6	-	8	623.6	1.1	80623.6	8	623.6	10.1	80521.3	8	521.3	0.2	0.00	0.13
D2_S4_C25_0	70683.1	-	7	683.1	62.9	70683.2	7	683.2	8.5	70629	7	629.0	0.2	0.00	0.08
D2_S4_C25_1	70601.0	-	7	601	3165.6	70601.0	7	601.0	10.0	60518.5	6	518.5	0.1	0.00	16.66
D2_S4_C25_2	70769.4	-	7	769.4	4.5	70769.4	7	769.4	9.6	70715.2	7	715.2	0.1	0.00	0.08
D2_S4_C25_3	70696.6	-	7	696.6	154.6	70701.8	7	701.8	8.5	70600.9	7	600.9	0.1	0.01	0.14
D2_S4_C25_4	80606.9	-	8	606.9	23.7	80610.3	8	610.3	9.8	80563.5	8	563.5	0.1	0.00	0.06
D2_S2_C30_0	90974.1	-	9	974.1	2.4	90980.7	9	980.7	13.2	90733.7	9	733.7	0.3	0.01	0.27
D2_S2_C30_1	70725.3	-	7	725.3	38.6	70725.3	7	725.3	13.2	60555.5	6	555.5	0.3	0.00	16.79
D2_S2_C30_2	81054.0	71374.4	8	1054	-	81083.1	8	1083.1	14.1	70824.8	7	824.8	0.2	0.04	14.48
D2_S2_C30_3	80898.6	-	8	898.6	232.3	80898.6	8	898.6	13.1	80673.8	8	673.8	0.1	0.00	0.28
D2_S2_C30_4	90758.0	-	9	758	7.8	90776.0	9	776.0	10.2	90634	9	634.0	0.2	0.02	0.16
D2_S4_C30_0	90841.8	80927.0	9	841.8	-	90847.9	9	847.9	13.7	90780.2	9	780.2	0.2	0.01	0.07
D2_S4_C30_1	70803.6	-	7	803.6	1107.1	70814.1	7	814.1	11.4	70604.2	7	604.2	0.2	0.01	0.30
D2_S4_C30_2	70904.3	-	7	904.3	107.9	70911.4	7	911.4	15.4	70808.4	7	808.4	0.1	0.01	0.15
D2_S4_C30_3	80838.8	80818.6	8	838.8	-	80838.8	8	838.8	12.2	80725.6	8	725.6	0.2	0.00	0.14
D2_S4_C30_4	100765.0	-	10	765	24.5	100810.1	10	810.1	9.1	100715	10	715.0	0.1	0.04	0.09

Table 6: Results on E-VSP instances containing 25 to 30 trips.

5.4. Results on the large instances

The ALNS solutions to the large instances with up to 8 depots, 16 stations and 500 trips are presented in Tab. 7. For each instance, we present the best solutions out of five random runs, the average solution of five random runs and the computational time in seconds for the algorithm where post optimization phase are presented . The last column shows the relative changes of the objective values when the post optimization is include. As can be seen from the table, the post optimization helps to improve the solution values by 3.1% on average.

A comparison between the E-VSP solution and the MD-VSP lower bound is given in Tab. 8. For the E-VSP solutions, we present the statistics for the best solution found in five random runs, including the solution value (Best sol.), the number of used vehicles (m), the total deadheading distance (d), the average per trip deadheading distance ($\frac{d^{LB}}{\#trips}$), the total number of recharges (r), the total number of partial recharges (r_p), the average remaining battery level before recharge (f_a) and the average remaining battery level after recharge (f_b). For the MD-VSP solutions, we present the optimal solution value (Opt sol.), the number of used vehicles (m), the total deadheading distance (d), the average per trip deadheading distance ($\frac{d^{LB}}{\#trips}$). The gaps between the two solutions are provided in the last column, calculated as $\frac{z_{E-VSP}-z_{MD-VSP}}{z_{MD-VSP}} \cdot 100$. The average values over each instance size are also given.

In the E-VSP solutions, the instances with 100 trips use 19.4 vehicles on average. Each vehicle takes 5.1 trips and recharges 1.5 times on average. For the instances with 500 trips, each vehicle carries out slightly more trips, 5.6, and also visits the recharging stations more often, 2.2 times, on average. Among all the recharging operations, nearly half of them are partial charging.

Comparing the E-VSP solutions with the MD-VSP lower bound, we can see that, for most of the instances with 100 trips, the E-VSP solution uses the same number of vehicles as the MD-VSP lower bound. The gap in the objective value is consistently below 7%. The total deadheading distance, is on average around 30% longer in the E-VSP solution than in the MD-VSP solution. This relatively high percentage is within our expectation because the total deadheading distance is only treated as the secondary objective, and because the deadheading in the E-VSP consists of not only the deadheading between the timetabled trips but also the deadheading to and from the recharging stations due to the highly constrained driving range.

As the number of trips increases from 100 to 500, the difference in the total deadheading distance between the two solutions increases significantly. This is because it becomes easier to find a more compact schedule for the MD-VSP when the number of trips is very high. For example, the average deadheading per trip (column $\frac{d^{LB}}{\#trips}$) in the MD-VSP solution is decreased from 13.6 km to 9.2 km when the number of trips is increased from 100 to 500 in the D4_S8_CX instances. The results also show that the E-VSP solution can be improved by providing more depots and stations. For a fixed number of trips, the difference between the E-VSP solution and the MD-VSP solution decreases as the number of stations increases.

In the E-VSP, the tendency of the average trip deadheading is hard to predict due to the restricted driving range, and heavily depends on the location of the stations. Fig. 3 depicts two different cases given by instance D4_S8_C500_2 and D4_S8_C500_5. In the former instance, the set of stations and depots are relatively well-spread, and the number of electric vehicles needed in the E-VSP is almost the same as that in the MD-VSP; whereas, in the latter instance, almost all the stations and depots are located in the left half of the plane, and moreover, four out of eight are clustered in the left bottom corner.

To further investigate how the distribution of the recharging station affects the solution, we have tested instance D4_S8_C500_2 and D4_S8_C500_5 with four stations distributed in four different ways: 1) even distribution, where the plane is divided into four equal sub-regions and one station is located in each sub-region; 2) random distribution, where the stations are randomly distributed in the plane; c) corner distribution, where the four stations are located at the four corners of the plane; d) quarter distribution, where the plane is divided into four equal sub-regions and four stations are located evenly within one sub-region. These distributions are illustrated in Fig. 4. The results are given in Tab. 9. For each instance and each station distribution, five random runs are performed. The average solution value, average number of used vehicles and average deadheading distances are presented in the table. Even distributed stations leads to the best solution as expected. The largest difference between two different distributions is more than 10%, as given by the even distribution and corner distribution for instance D4_S8_C500_5.

5.5. Sensitivity to the driving range and charging rate

To show the effect of driving range and charging rate on the E-VSP solution, we have tested the instance D2_S4_C100_1 with different battery capacities ranging from 135 km to 300 km, and with different charging rates from 0.2 minute to 2.4 minute charging per unit distance. The results are shown in Fig.5. As can be seen from the two plots in the top of Fig.5, the deadheading distance is reduced by nearly 30% when the driving range is increased from 135 km to 300 km, which shows that the driving range has a large effect on the total deadheading distance. The total number of performed charging operations as well as the number of performed partial charging operations reduces when the driving range approaches to 300 km.

Table 7: ALNS results on the large E-VSP instances.

	with Post Opt.			without Post Opt.		
Instance	best solution	Average solution	T (sec)	Average solution	T (sec)	Gap (%)
D2_S4_C100_1	211775	211784	252	211962	89	-0.1
D2_S4_C100_2	182178	188035	890	205957	102	-8.7
D2_S4_C100_3	192230	192270	998	212195	103	-9.4
D2_S4_C100_4	212231	212243	319	232322	107	-8.6
D2_S4_C100_5	181882	181892	381	201919	97	-9.9
D4_S8_C100_1	191600	191615	281	201770	91	-5.0
D4_S8_C100_2	192097	192112	500	202283	104	-5.0
D4_S8_C100_3	191510	191514	325	193863	101	-1.2
D4_S8_C100_4	211612	211629	306	221771	105	-4.6
D4_S8_C100_5	191704	191714	311	191983	105	-0.1
D4_S8_C500_1	878650	883592	1233	894326	483	-1.2
D4_S8_C500_2	940142	946231	1256	946231	506	0.0
D4_S8_C500_3	859788	869651	1187	878565	437	-1.0
D4_S8_C500_4	870033	883185	1187	888664	437	-0.6
D4_S8_C500_5	880386	898437	1198	922678	448	-2.6
D8_S16_C500_1	869530	881216	1255	890213	505	-1.0
D8_S16_C500_2	869282	877456	1259	886016	509	-1.0
D8_S16_C500_3	877456	877657	1178	881609	428	-0.4
D8_S16_C500_4	828538	839913	1189	847429	439	-0.9
D8_S16_C500_5	858816	874741	1183	885909	433	-1.3
Average	534572.1	539844.4	834.4	549883.3	281.4	-3.1

The charging rate also affects the solution significantly as shown in the two plots in the bottom of Fig. 5. Before the charging rate reaches 2 min/km, the number of used vehicles remains the same and the total deadheading distance increases. When it reaches 2 min/km, the total deadheading distance drops due to the use of an extra vehicle. The objective value consistently increases as the charging rate increases. The number of charging operations increases in general as the charging rate increases. The number of partial charging also increases because it takes long to charge fully when the charging rate is high.

6. Conclusion and future work

In this paper, we have considered a new vehicle scheduling problem, the Electric Vehicle Scheduling Problem with Partial Charging, in which a set of electric vehicles with limited driving range are scheduled to perform a set of timetabled trips and allowed to be recharged fully or partially at any given recharging station. The charging time is assumed to be a linear function of the charged amount. We have presented a mixed integer programming formulation for the problem, and developed an ALNS heuristic to solve the problem. In ALNS, a wide range of repair- and destroy-methods are adopted, a diversification phase is applied occasionally, and a post optimization phase is implemented to further improve the solution.

This work can be extended in a few directions in the future. Firstly, a better lower bound for the E-VSP can be investigated, which is not trivial due to a high complexity introduced by the partial charging. Secondly, the linear function used in this work could be replaced by a more realistic function to better describe the relation among charging time, charged amount and remaining battery level. Thirdly, it would be very helpful to study a combined recharging station location and vehicle scheduling problem in order to determine the locations of the recharging stations at a strategic planning level. It is also interesting to investigate how much operational cost and how many electric vehicles can be saved by allowing slight modifications of the timetables in the E-VSP.

Table 8: Comparison between E-VSP and MD-VSP on the large E-VSP instances.

	E-VSP								MD-VSP				
Instance	Best sol.	m	d	$\frac{d}{\#trips}$	r	r_p	f_a	f_b	Opt sol.	m	d	$\frac{d}{\#trips}$	Gap (%)
D2_S4_C100_1	211775	21	1775	17.8	19	5	0.33	0.98	211460	21	1460	14.6	0.1
D2_S4_C100_2	182178	18	2178	21.8	38	23	0.28	0.83	171530	17	1530	15	6.2
D2_S4_C100_3	192230	19	2230	22.3	33	17	0.34	0.94	181637	18	1637	16.4	5.8
D2_S4_C100_4	212231	21	2231	22.3	31	7	0.32	0.88	211550	21	1550	15.5	0.3
D2_S4_C100_5	181882	18	1882	18.8	33	14	0.33	0.86	181368	18	1368	13.7	0.3
Average	196059	19.4	2059	20.6	30.8	13.2	0.32	0.90	191509	19	1509	15.1	2.4
D4_S8_C100_1	191600	19	1600	16.0	28	13	0.33	0.90	191270	19	1270	12.7	0.2
D4_S8_C100_2	192097	19	2097	21.0	37	11	0.34	0.93	191694	19	1694	17	0.2
D4_S8_C100_3	191510	19	1510	15.1	24	10	0.41	0.96	191209	19	1209	12.1	0.2
D4_S8_C100_4	211612	21	1612	16.1	28	9	0.42	0.93	211245	21	1245	12.5	0.2
D4_S8_C100_5	191704	19	1704	17.0	25	5	0.35	0.93	191375	19	1375	13.8	0.2
Average	195705	19.4	1705	17.0	28.4	9.6	0.37	0.93	195358.6	19.4	1358.6	13.6	0.2
D4_S8_C500_1	878650	87	8650	17.3	185	101	0.34	0.79	834311	83	4311	8.6	5.3
D4_S8_C500_2	940142	93	10142	20.3	185	120	0.23	0.76	924742	92	4742	9.5	1.7
D4_S8_C500_3	859788	85	9788	19.6	194	74	0.38	0.89	764554	76	4554	9.1	12.5
D4_S8_C500_4	870033	86	10033	20.1	186	77	0.37	0.92	824795	82	4795	9.6	5.5
D4_S8_C500_5	880386	87	10386	20.8	196	84	0.36	0.91	784672	78	4672	9.3	12.2
Average	885800	87.6	9800	19.6	189.2	91.2	0.34	0.85	826614.8	82.2	4614.8	9.2	7.2
D8_S16_C500_1	869530	86	9530	19.1	242	109	0.44	0.86	814162	81	4162	8.3	6.8
D8_S16_C500_2	869282	86	9282	18.6	188	61	0.39	0.88	824310	82	4310	8.6	5.5
D8_S16_C500_3	877456	87	7456	14.9	167	71	0.38	0.82	874279	87	4279	8.5	0.4
D8_S16_C500_4	828538	82	8538	17.1	195	106	0.37	0.88	783862	78	3862	7.7	5.7
D8_S16_C500_5	858816	85	8816	17.6	191	85	0.35	0.88	824290	82	4290	8.6	4.2
Average	860725	85.2	8725	17.4	196.6	86.4	0.38	0.86	824180.6	82	4180.6	8.4	4.4

Table 9: Results of different location distribution scenarios.

	Instance D4_S8_C500_2			Instance D4_S8_C500_5		
Station distribution	solution value	m	d	solution value	m	d
Even distribution	948710.7	93.8	10710.7	896651.8	88.6	10651.8
Random distribution	957596.5	94.8	9596.5	935726.4	92.4	11726.4
Corner distribution	999205.4	98.6	13205.4	986614.1	97.4	12614.1
Quarter distribution	986029.7	97.4	12029.7	945256.5	93.4	11256.5

Acknowledgments

This work is a part of the SELECT (Suitable ELEctromobility for Commercial Transport) project funded by the Danish Strategic Research Council.This support is gratefully acknowledged. The authors also thank the two anonymous referees for their valuable comments that contributed to improve the quality of the paper.

Bibliography

J. D. Adler. *Routing and Scheduling of Electric and Alternative-Fuel Vehicles*. PhD thesis, Arizona State University, 2014.

J. J. Bakker. Contesting range anxiety: The role of electric vehicle charging infrastructure in the transportation transition. Master's thesis, Eindhoven University of Technology, 2011.

L. Bodin, D. Rosenfield, and A. Kydes. Ucost: a micro approach to a transportation planning problem. *Journal of Urban Analysis*, 5 (1):47–69, 1978.

L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research*, 10(2):63–211, 1983.

S. Bunte and N. Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1(4):299–317, 2009.

G. Carpaneto, M. Dell'Amico, M. Fischetti, and P. Toth. A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19(5):531–548, 1989.

G. Desaulniers, F. Errico, S. Irnich, and M. Schneider. Exact algoirthm for electric vehicle routing problems with time window. Technical report, Les Cahiers du GERAD, 2014. URL https://www.gerad.ca/en/papers/G-2014-110/view.

Martin Desrochers and Gilbert Laporte. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado. 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, 2014.

R. Freling and J. M. P. Paixao. Vehicle scheduling with time constraint. In *Computer-Aided Transit Scheduling*, pages 130–144. Springer, 1995.

D. Goeke and M. Schneider. Routing a Mixed Fleet of Electric and Conventional Vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015.

A. Haghani and M. Banihashemi. Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. *Transportation Research Part A: Policy and Practice*, 36(4):309–333, 2002.

G. Hiermann, J. Puchinger, S. Ropke, and R.F. Hartl. The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. *European Journal of Operational Research*, In Press, Accepted Manuscript, 2016.

N. Kliewer, V. Gintner, and L. Suhl. Line change considerations within a time-space network based multi-depot bus scheduling model. In *Computer-aided Systems in Public Transport*, pages 57–70. Springer, 2008.

J. K. Lenstra and A. H. G. R. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

Movia. Miljoregnskab 2014. Accessed on Feb 11, 2016. URL https://www.moviatrafik.dk/media/4579/miljoerapport_2014.pdf.

A.-S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, 12(1):17–30, 2009.

J.-Y. Potvin and J.-M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340, 1993.

S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.

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

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming–CP98*, pages 417–431. Springer, 1998.

World Health Organization. Burden of disease from household air pollution for 2012. Accessed on Feb 11, 2016. URL http://www.who.int/phe/health_topics/outdoorair/databases/FINAL_HAP_AAP_BoD_24March2014.pdf.

Figure 3: The locations of the stations (rectangle and triangle), the depots (rectangle), and trip start and end points (circle) in instance D4_S8_C500_2 (left) and D4_S8_C500_5 (right).

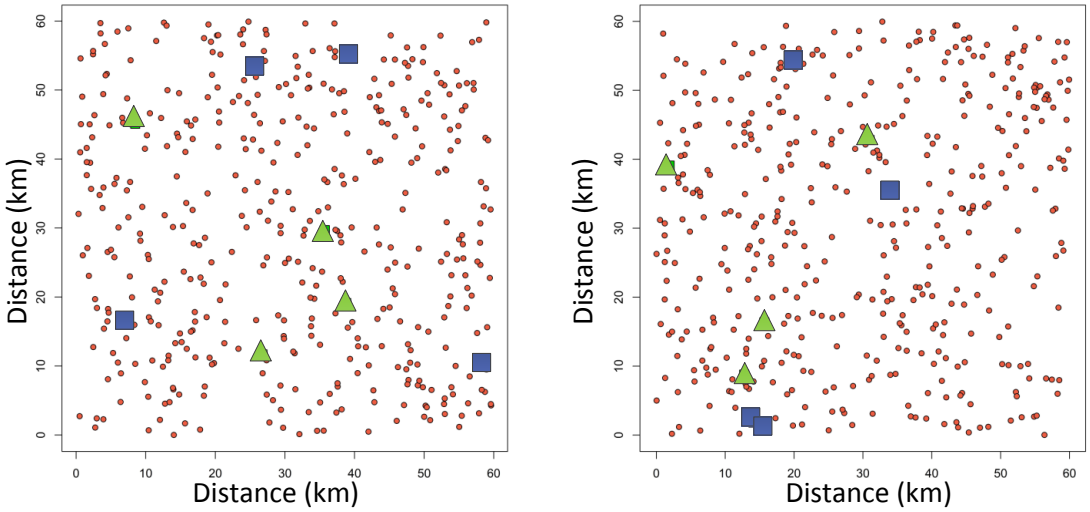


Figure 4: Four different types of station locations.

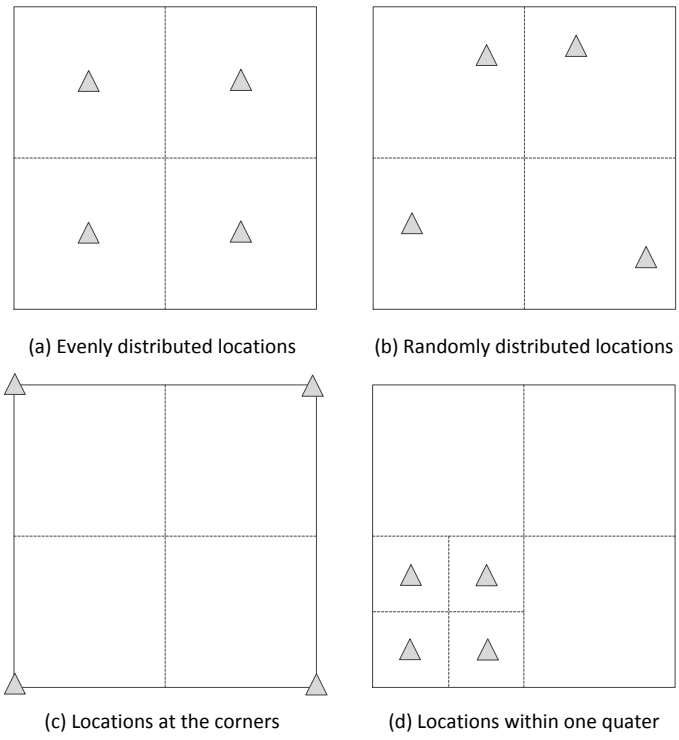


Figure 5: Sensitivity to the driving range and charging rate.