Available online at www.sciencedirect.com
ScienceDirect
IFAC Papers Online
CONFERENCE PAPER ARCHIVE
IFAC-PapersOnLine 49-12 (2016) 955–960

A Hybrid Iterative Local Search Algorithm for The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations

Puca Huachi Vaz Penna * **H.Murat Afsar** ** **Christian Prins** **
Caroline Prodhon **

* *Universidade Federal de Ouro Preto - Departamento de Computação, Campus Morro do Cruzeiro, s/n, Ouro Preto-MG, 35400-000, Brazil*
** *ICD-LOSI, UMR CNRS 6281, Université de Technologie de Troyes, 12 Marie Currie, CS 42060, 10004 Troyes CEDEX, France.*

Abstract: As the cities around the world become larger, quality of life of the citizens is more and more threatened due to the traffic congestion, energy consumption, noise disturbance and carbon emissions because of of increasing transport. Electrical vehicles present an opportunity to reduce greenhouse gas emissions. But limited driving range and long recharging times are among the challenges that the research community has to face. This paper proposes an iterative local search algorithm coupled with a set partitioning model to solve The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. Several types of electrical vehicles with varying driving range, capacity and fixed cost should service a set of customers within their time limits and during their tours each vehicle can be recharged in stations. We show the efficiency and the quality of our method by solving benchmark instances of the Heterogeneous Fleet Electric Vehicle Problems with Time Windows.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Electrical vehicles, Heterogeneous fleet, Metaheuristic

1. INTRODUCTION

In the modern society, reduction of the carbon dioxide emission is a major challenge. Annual European Union greenhouse gas inventory state that total carbon dioxide (CO_2) emission due to road transportation increases for 123 million ton between 1990 and 2012 for 28 member countries of European Union. Between 2011 and 2012 a 4% decrease is observed because of persisting economic downturn or recession (Agency, 2014). Road transport contributes about one-fifth of the EU's total emissions of CO_2, the main greenhouse gas. Trucks, buses and coaches produce about a quarter of CO_2 emissions from road transport in the EU and some 5% of the EU's total greenhouse gas emissions – a greater share than international aviation or shipping (EU, 2014). Reducing CO_2 emission requires efforts to limit fossil fuels and promote cleaner energy sources such as natural gas, ethanol, or electricity.

Electric vehicles contribute to diminishing the CO_2 emissions of transportation activities especially in urban areas. Electric vehicles (EV) powered by batteries can be charged by regular power outlets and offer lower operating costs than combustion engine cars. They are also efficient, emit no pollution, and are almost noiseless. However, all-electric vehicles also come along with several disadvantages. Batteries only have a certain operating distance and Recharging batteries currently takes too much time (Eggers and Eggers, 2011).

The cost competitiveness of electric trucks compared to conventional trucks influences the large-scale implementation of EVs. Case study companies calculated the total costs of ownership (including costs of investment for EV and charging infrastructure, costs for energy and other costs such as vehicle tax, insurance, service and maintenance, repairs and environmental charges) as key financial indicator for profitability(Taefi et al., 2014). In that context, the fleet management is extremely important where the capacity, driving range and fixed cost of each type of vehicle are different. We propose an hybrid iterative local search algorithm with an embedded set partitioning model to solve The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations.

In the reminder of this paper we will present the relevant literature in section 2. The section 3 will define the problem and we will give a comprehensive description of the solution algorithm in section 4. The relevance of the this approach will be justified in section 5 and section 6 will conclude this paper.

2. STATE OF THE ART

Even though minimizing the fuel consumption through alternative objective functions is studied in papers like Kara et al. (2007), recharging stations have started to be considered in routing problems only very recently. Erdoğan and Miller-Hooks (2012) define a Green Vehicle Routing Problem (G-VRP) where a vehicle can extend its driving range by visiting recharging stations. They are inspired by

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.899

the VRP with Satellite Facilities (VRPSF) proposed by Bard et al. (1998). Wang (2008) uses integer programming to optimize the location and number of battery exchange stations for electric scooters. More recently in 2013 they present a set and maximum coverage model to maximize the coverage of EV flows on paths by the siting of multiple types of stations (Wang and Lin, 2013). He et al. (2013) develop an equilibrium modelling framework applied to determine an optimal allocation of a given number of public charging stations among metropolitan areas in the region to maximize social welfare associated with the coupled networks.

Sweda and Klabjan (2012) are interested in finding a minimum-cost path for an EV when the vehicle must recharge along the way and they solve this problem by two methods based on dynamic programming. Gonçalves et al. (2011) solve a VRP with pick up and delivery using a heterogeneous fleet of vehicles using fossil fuel and electrical motors. RG and MA (2011) define a Recharging Vehicle Routing Problem where electrical vehicles can be recharged in customer locations.

Schneider et al. (2014) offer the Electric Vehicle Routing Problem with Time Windows and Recharging Stations. In this problem, each arc causes the vehicle to consume a certain amount of energy, proportional to its length. They find capacity and charge feasible routes servicing each customer in a given time window. The vehicles are identical, having the same capacity and charge level. Visiting a recharging node will extend the driving span of the vehicle by recharging the difference between the present charge level and the battery capacity, with a recharging rate i.e., the recharging time depends on the energy level of the vehicle when arriving at the respective station. They solve it with a hybrid VNS/Tabu search. Goeke and Schneider (2015) extend the former work by considering conventional and electrical vehicles in the same fleet. Moreover, they study the effects of vehicle speed and load in fuel/energy consumption. This problem is solved by an Adaptive Large Neighbourhood Search algorithm. Sassi et al. (2014) study a very similar problem in which they incorporate time dependent recharging costs.

Finally Hiermann et al. (2014) include several types of electrical vehicles which differ in capacity, charge level and fixed cost. The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations (E-FSMVRPTW). An exact method by Branch and Price and a metaheuristic based on an Adaptive Large Neighbourhood Search method solve this problem on instances of Schneider et al. (2014), modified according to Liu and Shen (1999).

3. PROBLEM DEFINITION

The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations (E-FSMVRPTW) can be defined as follows. Let $G = (N, A)$ be a directed graph where $N = V \cup F$ and $A = \{(i, j) : i, j \in N, i \neq j\}$ is the set of arcs. For each arc, a distance d_{ij} and a travel time t_{ij} are associated. The set $V = \{0, 1, \ldots, n\}$ is composed by $n+1$ vertices and the vertex 0 denotes the depot, where the vehicle fleet is located, while the set $V' = V \setminus \{0\}$ is composed by the remaining vertices that represents the n customers. $F = F' \cup F_0$, where $F' = \{1, \ldots, n'\}$ is a set of n' recharging stations and F_0 a copy of the depot 0, which can also be a recharging station. Each customer $i \in V'$ has a non-negative demand q_i, a service time s_i and must be attended inside the interval $[e_i, l_i]$, where e_i and l_i define, respectively, the start and the end of time windows. The fleet is composed by m different types of vehicles, with $M = \{1, \ldots, m\}$. Every vehicle type $k \in M$ is associated with a load capacity Q_k and battery capacity Y_k. The battery energy consumption rate per distance unit for each vehicle type is defined by r_k and the battery recharging rate is g_k. With the vehicle is also associated a fixed cost denoted by f_k. For each arc $(i, j) \in A$ there is an associated energy consumption $c_{ij}^k = d_{ij} \times r_k$. Finally, the time to recharge the battery defines the service time s_h of a recharging station h and it is given by $s_h = y_h \times g_k$, where y_h is the consumed energy, of the vehicle k, while arriving at the station h. The recharging time is assumed to be linear as in Schneider et al. (2014).

The objective is to determine the best fleet composition as well as the set of routes, $R^k = (i_1, i_2, \ldots, i_{|R|})$, that minimize the sum of fixed and travel costs in such a way that: (i) every route R^k starts and ends at the depot ($i_1 = i_{|R|} = 0$ and $\{i_2, \ldots, i_{|R|-1} \subseteq V' \cup F\}$) and is associated to a vehicle type k ($k \in M$); (ii) each customer belongs to exactly one route; (iii) each recharging station can be visited more than once, in the same route or not, and also consecutively; (iv) the load capacity Q_k of a vehicle type k is not exceeded ($\sum_{j=2}^{|R|-1} q_{i_j} \leqslant Q_k$). (v) the vehicle autonomy travel distance must respect the energy battery capacity.

4. SOLUTION APPROACH

Here we describe the heuristic algorithm proposed to solve the E-FSMVRPTW. This algorithm is an extension of the algorithm presented by Subramanian et al. (2012) and Subramanian et al. (2013). The proposed hybrid heuristic, named HILS, is a multi-start heuristic composed by an ILS (Lourenço et al., 2010) metaheuristic and a Set Partitioning (SP) formulation well integrated.

According to Lourenço et al. (2010) the ILS is a memory-less metaheuristic, i.e., the ILS does not use any information, previously obtained, to improve the solution quality. Our goal is to use the SP formulation, embedded in the multi-start ILS framework, as a mechanism to add memory to it. Thereby, obtaining a robust algorithm to solve the E-FSMVRPTW. The memory mechanism is based on creating a set of good quality routes, named Route Pool ($\mathcal{RP}$), during the ILS phase. At the end of the each iteration of the multi-start phase, a SP model is generated, using the $\mathcal{RP}$, and a Mixed Integer Programming (MIP) solver is called to solve SP model and obtain the problem solution.

Let $\mathcal{R}$ be the set of all possible routes of all vehicle types, $\mathcal{R}_i \subseteq \mathcal{R}$ be the subset of routes that contain customer $i \in V'$. Define λ_k as the binary variable associated to a route $k \in \mathcal{R}$, and c_k as its cost. The SP formulation for the E-FSMVRPTW can be expressed as follows.

$$\text{Min} \sum_{k \in \mathcal{R}} c_k \lambda_k \tag{1}$$

subject to

$$\sum_{k \in \mathcal{R}_i} \lambda_k = 1 \qquad \forall i \in V' \tag{2}$$

$$\lambda_k \in \{0, 1\}. \tag{3}$$

The objective function (1) minimizes the sum of the costs by choosing the best combination of routes. Constraints (2) state that a single route from the subset $\mathcal{R}_i$ visits costumer $i \in V'$. Constraints (3) define the domain of the decision variables. Since this complete formulation has an exponential number of variables, it can not be directly solved. Solving it by branch-and-price or related methods, as done in some proposed exact algorithms, is time-consuming and only practical up to a certain instance size. The HILS algorithm actually solves a SP problem similar to (1–3), where $\mathcal{R}$ is restricted to the $\mathcal{RP}$, generated by the ILS based metaheuristic, named ILS-RVND, with a few thousands routes. It is important to mention, that the recharging stations can be visited as many times as necessary. Thus, they are not included in the SP model.

Here we present a high-level description of the HILS algorithm. For a full description and implementation details refer to Subramanian et al. (2012) and Subramanian et al. (2013). The general HILS is given bellow.

Step 1. Read problem data, initialize the number of multi-start iterations ($MSiter$) and the $\mathcal{RP}$.
Step 2. Run the ILS-RVND without given an initial solution.
Step 3. Update the $\mathcal{RP}$
 (a) Add all the routes, that are not yet in the $\mathcal{RP}$, associated to best ILS-RVND solution. These routes are called permanent routes and they define a long-term memory.
Step 4. Create the SP model using the $\mathcal{RP}$.
Step 5. Call the MIP solver to solve the SP model.
 (a) Whenever a incumbent solutions is found, run the ILS-RVND with the incumbent solution given to the ILS-RVND, as the initial solution.
 (b) If a solution is found in the root node, increase the *tolerance* value, that controls the $\mathcal{RP}$ size.
 (c) If the time limit given to the MIP solver is exceed, decrease the *tolerance* value and remove all non-permanent routes given by the ILS-RVND.
Step 6. If the MIP solver or the ILS-RVND improve the current solution
 (a) update the best solution.
Step 7. if $MSiter$ is less than the maximum number of multi-start iterations go to Step 2.
Step 8. Return the best solution obtained.

It is important to highlight, that the MIP solver calls the ILS-RVND heuristic whenever an incumbent solution is found. Despite of the $\mathcal{RP}$ had being generated by the ILS-RVND, this intensive search is important. Maybe the ILS-RVND has not visited a solution composed by that subset of routes found by the MIP solver, when solving the SP model. We assume that the MIP solver uses a Branch-and-bound or a Branch-and-cut solution procedure and its stopping criteria are: (i) optimal solution found; (ii) the execution time of the solver is greater than a time limit given parameter.

4.1 The ILS-RVND heuristic

The ILS-RVND heuristic is an extension of the one developed by Penna et al. (2013) for the Heterogeneous Fleet VRP. However, several modifications were done to tackle the E-FSMVRPTW.

The ILS metaheuristic includes four main components: (i) initial solution construction; (ii) local search; (iii) perturbation; (iv) acception criteria. In order to maintain ILS simplicity, our ILS-RVND implementation uses the same components. The general ILS-RVND is described below (we refer to Penna et al. (2013) for a detailed description).

Step 1. Initialize the number of ILS iterations ($ILSiter$).
Step 2. If an initial solution is not given, generate the initial solution, using a constructive procedure.
Step 3. Run the local search
 (a) Update the time warp and the auxiliary data structures (ADS).
 (b) Initialize the list of inter-route neighborhoods
 (c) Choose a neighborhood at random and find the best neighbor of current solution
 (d) If the best neighbor improves the current solution
  (i) update the current solution.
  (ii) Run the Intra-route search.
  (iii) Restart the list of inter-route neighborhoods.
 (e) If the best neighbor does not improves the current solution, remove the selected neighborhood from the list.
 (f) Update the ADS and fleet and remove extra consecutive recharging stations.
Step 4. Update the $\mathcal{RP}$
 (a) if the current solution cost is less than the best ILS solution times a percent *tolerance* value, add all the current solution routes, that are not yet in the $\mathcal{RP}$. These routes are called non-permanent routes and they define a short-term memory.
Step 5. If the local search improves the current solution.
 (a) update the best solution.
 (b) restart $ILSiter$
Step 6. Run the perturbation procedure.
Step 7. if $ILSiter$ is less than the maximum number of ILS iterations go to Step 3.
Step 8. Return the best ILS solution and the $\mathcal{RP}$.

The main ILS-RVND components and some features are explained below.

Constructive Procedure The constructive procedure uses a simple greedy method with a random start point. Initially, for each given vehicle type, one empty route is created and filled with a visiting point randomly selected. This visiting point can be a customer or a recharging station. Next, the routes are filled using one of the two insertion criteria defined. The first, consists of an adaptation of the classical Nearest Insertion Criterion. The second consists of a modification of the well-known Cheapest Insertion Criterion including an insertion incentive to visiting points located far from the depot be inserted firstly. Thus, the number of vehicles is unlimited, it is always possible to add one more vehicle (randomly selected) to the initial solution, whenever there is no available capacity to insert the customers. In the same way, if there is no remaining energy capacity and continue to exist customers out of the solution, some recharging stations are randomly selected and inserted. This process is repeated until all customers and recharging stations are inserted. After that, one empty route associated to each vehicle type is created. This allow a possible fleet resizing during the local search phase. We do not admit vehicle loading capacity neither vehicle energy capacity violations. Nevertheless, we ignore the time windows constraint during the construction phase. The time windows violation is penalized using time warp structure (Vidal et al., 2013).

Local Search The local search uses a Randomized VND (RVND) procedure. Five inter-route neighbourhoods are implemented based on λ-interchange scheme (Osman, 1993), which consists of exchanging up to λ visiting points between two routes. To limit the number of possibilities we have considered $\lambda = 2$. These moves are the same used by Penna et al. (2013) for addressing heterogeneous fleet. Whenever, a move improves the current solution an intra-route local search is performed. It is composed by five well-known intra-route neighborhood structures: reinsertion, Or-opt (with 2 and 3 adjacent visiting points), 2-opt and exchange moves. The solution spaces of all defined neighborhoods are explored exhaustively, i.e., all possible combinations are examined, and the best improvement strategy is considered. During the local search only feasible moves, related to load and energy constraints, are admitted. Thus, the initial solution admits time windows violation, the local search accept moves that minimize the time windows violation and once the solution becomes feasible no more violation is admitted. The move evaluation process can be done in constant time for the load and energy constraints. Since, we use a time warp structure for dealing with the time windows constraints, it also allows the move evaluating in constant time (Vidal et al., 2013). Although, a pseudo-polynomial time is necessary to update the time warp structure. As defined by Penna et al. (2013), some Auxiliary Data Structure (ADS) are adopted, in order to enhance the neighborhood search.

Perturbation The perturbation is used as a diversification mechanism based on multiple swap moves. Four perturbation mechanisms were adopted. Three of them are the same moves implemented by Penna et al. (2013): Multiple-Swap(1,1), Multiple-Shift(1,1) and Split. A new perturbation is introduced, named AddRechargingStations. This new perturbation adds h recharging stations to the solution, where h is randomly selected from the interval $\{1, \ldots, |F|\}$. In the AddRechargingStations perturbation, the recharging station to be added, the route where it will be inserted and its position are randomly selected. Like the constructive procedure and local search, these four perturbations accept only feasible moves, except for time windows constraint.

5. NUMERICAL RESULTS

The algorithm HILS was coded in C++ (g++ 4.8.2) and executed in an Intel Core i7 Processor 2.93 GHz with 8 GB of RAM running Ubuntu Linux 14.04 (Kernel 3.10 – 64 bits). The SP formulation was implemented using the solver CPLEX 12.5.1. The developed approach was tested with the instances proposed by Hiermann et al. (2014). This benchmark set consists in 168 instances, divided in several subsets according to: (i) customers distribution, clustered (C), randomly distributed (R) or a combination of both (RC); (ii) scheduling horizon, shorter (1) or longer (2); (iii) vehicle costs, large (a), medium (b) or small (c), as defined by Liu and Shen (1999).

The HILS was executed 10 times for each instance and the number of multi-start iterations of the algorithm parameter was set to 50. The others are equal to the parameter setting presented by Subramanian et al. (2012).

The computational results obtained by the HILS are compared to the results obtained by the ALNS proposed by Hiermann et al. (2014). They are summarized in Table 1. In this table, the first column, **Inst.**, denotes the benchmark subset name and the total number of instances. For each algorithm, **Sol.** indicates the average solution cost for the subset, **Gap** corresponds to the average gap between the best solutions found by the algorithm and the best known solution and **Time** indicates the computational time in seconds. The **BKS** corresponds to the number of solutions where HILS equal (=) or improve (<) the best-known solutions (BKS), for each benchmark subset. Finally, the rows **Min. gap**, **Agv. gap** and **Max. gap** denote the minimum, average and maximum gap for all instances, respectively. A negative gap indicates that the algorithm was able to improve the BKS. The **# Sol. found** is the total number of BKS obtained. The **Avg. Time** row presents the average running time in seconds and **CPU** is the processor used by each algorithm.

The results presented on Table 1 show that the HILS was able to obtain 76 (45.24%) new best-known solutions and equal 8 (4.76%) solution. The average gap between best solutions obtained by the HILS and the BKS was only 0.23, where the maximum improvement was 1.22%. The HILS was more efficient on instances of the subset with vehicle cost type (a), i.e., when the vehicle costs are larger, and a small change in fleet composition may cause a relevant impact on final solution cost.

6. CONCLUSION

This paper dealt with The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations (E-FSMVRPTW). This problems is variant of the well-known Fleet Size and Mix VRP with time windows, where the fleet is composed by a set of heterogeneous electrical vehicles, i.e., with different capacities, costs and battery size.

The problem was solved by a multi-start hybrid algorithm based on the Iterated Local Search (ILS) metaheuristic

Table 1. HILS Computational Results

Inst.	$ALNS_{2000}$ Sol. cost	Gap (%)	Time (s)[1]	HILS Sol. cost	Gap (%)	Time (s)[2]	BKS =	BKS <
C1a (9)	7157.85	0.24	1060.60	7130.79	-0.14	580.46	0	9
C2a (8)	5732.89	0.39	2552.03	5706.55	-0.07	741.32	0	7
R1a (12)	4134.27	1.05	930.80	4064.67	-0.65	506.57	0	11
R2a (11)	3171.83	0.87	2754.05	3148.91	0.13	968.15	0	5
RC1a (8)	5049.05	1.26	895.28	4974.00	-0.24	642.41	0	6
RC2a (8)	4229.42	0.61	962.25	4198.83	-0.12	504.27	0	6
C1b (9)	2462.09	0.48	880.67	2449.55	-0.04	677.14	2	5
C2b (8)	1722.75	1.08	2223.15	1709.22	0.29	737.58	1	2
R1b (12)	1910.27	1.66	911.10	1882.53	0.19	581.24	0	2
R2b (11)	1353.35	1.28	1768.64	1357.21	1.51	943.34	0	0
RC1b (8)	2203.12	1.38	874.13	2174.66	0.09	756.11	0	3
RC2b (8)	1707.93	1.44	1131.98	1688.86	0.31	585.22	1	0
C1c (9)	1769.72	0.44	886.33	1762.86	0.05	655.60	0	2
C2c (8)	1204.07	0.89	1917.00	1204.15	0.89	699.01	2	0
R1c (12)	1588.86	1.81	930.30	1566.92	0.41	661.11	0	11
R2c (11)	1128.24	1.72	1723.15	1124.14	1.28	955.95	1	1
RC1c (8)	1815.86	1.51	888.23	1787.52	-0.10	755.98	0	5
RC2c (8)	1374.07	1.28	1182.53	1361.81	0.40	616.13	1	1
Min. gap		0.10			-1.22			
Agv. gap		1.08			0.23			
Max. gap		2.85			2.32			
# Sol. found							8	76
Agv. Time (s)			1359.57			698.20		
CPU	Core2 Quad 2.40GHz			i7 2.93GHz				

[1]: Original time in minutes

and a Set Partitioning (SP) formulation well integrated. The ILS was used to generate a set of good quality routes. Every multi-start iteration, a MIP solver was called to solve the SP model generated using the set of routes.

The proposed algorithm (HILS) was tested on 168 benchmark instances with 100 customers and it was capable to obtain 76 new improved solutions and to equal the result of 8 instances. Thus, the total of best-known solutions (BKS) obtained by the HILS was 50% of the total number of instances. The average gap between the solutions obtained by HILS and the BKSs was only 0.23%. This fact clearly illustrates the robustness in terms of solution quality of the hybrid approach.

ACKNOWLEDGEMENTS

This work was partially supported by CNPq – Brazil.

REFERENCES

Agency, E.E. (2014). Annual european union greenhouse gas inventory 1990–2012 and inventory report 2014. Technical report, UNFCCC Secretariat.

Bard, J.F., Huang, L., Dror, M., and Jaillet, P. (1998). A branch and cut algorithm for the vrp with satellite facilities. *IIE transactions*, 30(9), 821–834.

Eggers, F. and Eggers, F. (2011). Where have all the flowers gone? forecasting green trends in the automobile industry with a choice-based conjoint adoption model. *Technological Forecasting and Social Change*, 78(1), 51 – 62.

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

EU (2014). Road transport: Reducing CO2 emissions from vehicles. `http://ec.europa.eu/clima/policies/transport/vehicles/heavy/index_en.htm` . Accessed: 2015-06-18.

Goeke, D. and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1), 81 – 99.

Gonçalves, F., Cardoso, S.R., Relvas, S., and Barbosa-Póvoa, A. (2011). Optimization of a distribution network using electric vehicles: A vrp problem. *IO2011-15 Congresso da associação Portuguesa de Investigação Operacional*.

He, F., Wu, D., Yin, Y., and Guan, Y. (2013). Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 47(0), 87 – 101.

Hiermann, G., Puchinger, J., and Hartl, R.F. (2014). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Technical report, Working Paper. Accessed July 17, 2014. URL: http://prolog.univie.ac.at/research/publications/downloads/Hie_2014_638.pdf.

Kara, I., Kara, B., and Yetis, M. (2007). Energy minimizing vehicle routing problem. In A. Dress, Y. Xu, and B. Zhu (eds.), *Combinatorial Optimization and Applications*, volume 4616 of *Lecture Notes in Computer Science*, 62–71. Springer Berlin Heidelberg.

Liu, F.H. and Shen, S.Y. (1999). The fleet size and mix vehicle routing problem with time windows. *The Journal of the Operational Research Society*, 50(7), pp. 721–732.

Lourenço, H.R., Martin, O.C., and Stützle, T. (2010). Iterated local search: Framework and applications. In M. Gendreau and J.Y. Potvin (eds.), *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, 363–397. Springer US.

Osman, I.H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(1-4), 421–451.

Penna, P.H.V., Subramanian, A., and Ochi, L.S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2), 201–232.

RG, C. and MA, F. (2011). The recharging vehicle routing problem. In *Proc. 2011 Indust. Engrg. Res. Conf., Reno, CA.*

Sassi, O., Cherif, W.R., and Oulamara, A. (2014). Vehicle Routing Problem with Mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs.

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

Subramanian, A., Penna, P.H.V., Uchoa, E., and Ochi, L.S. (2012). A hybrid algorithm for the heterogenous fleet vehicle routing problem. *European Journal of Operational Research*, 221, 285 – 295.

Subramanian, A., Uchoa, E., and Ochi, L.S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10), 2519 – 2531.

Sweda, T.M. and Klabjan, D. (2012). Finding minimum-cost paths for electric vehicles. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, 1–4. IEEE.

Taefi, T., Kreutzfeldt, J., Held, T., Konings, R., Kotter, R., Lilley, S., Baster, H., Green, N., Laugesen, M.S., Jacobsson, S., Borgqvist, M., and Nyquist, C. (2014). A systematic case study approach with examples from denmark, germany, the netherlands, sweden and the uk.

Vidal, T., Crainic, T.G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1), 475 – 489.

Wang, Y.W. (2008). Locating battery exchange stations to serve tourism transport: A note. *Transportation Research Part D: Transport and Environment*, 13(3), 193 – 197.

Wang, Y.W. and Lin, C.C. (2013). Locating multiple types of recharging stations for battery-powered electric vehicle transport. *Transportation Research Part E: Logistics and Transportation Review*, 58(0), 76 – 87.