Appendix of: The Electric Vehicle Routing Problem with Time Windows and Recharging Stations

Michael Schneider∗ Andreas Stenger† Dominik Goeke∗
March 8, 2013

A Improved Problem Formulation for G-VRP

We compiled an improved version of the G-VRP model of Erdogan and Miller-Hooks (2012) and used it to solve the set of small G-VRP benchmark instances with CPLEX. To be concise, we use the original notation from Erdogan and Miller-Hooks (2012) to describe our model:

\( v_0 \) Depot

\( I \) Set of customers

\( I_0 \) Set of customers and the depot, \( I_0 = I \cup \{v_0\} \)

\( F \) Set of AFS vertices

\( F' \) Set of visits, which are dummy vertices created from \( F \)

\( F_0 \) Set of visits and the depot, \( F_0 = F' \cup \{v_0\} \)

\( V \) Set of real vertices, \( V = \{v_0\} \cup I \cup F \)

\( V' \) Set of vertices, including dummy vertices, \( V' = \{v_0\} \cup I \cup F' \)

\( Q \) Vehicle fuel capacity

\( r \) Vehicle fuel consumption rate

\( y_j \) Variable specifying the remaining fuel level when reaching vertex \( j \), is set to \( Q \) at each AFS visit and the depot

\( T_{\text{max}} \) Maximal duration of each route

\( t_{ij} \) Travel time between vertices \( i \) and \( j \)

\( p_i \) Service time at vertex \( i \), (if \( i \in I \), then \( p_i \) is service time of the customer vertex, if \( i \in F' \), then \( p_i \) is the refueling time at the AFS vertex

\( \tau_j \) Variable specifying the time of arrival (before start of service) at vertex \( j \)

\( x_{ij} \) Binary decision variable, equal to 1 if a vehicle travels from vertex \( i \) to \( j \), 0 otherwise

∗{schneider,goeke}@wiwi.uni-kl.de, Chair of Business Information Systems and Operations Research, Technical University Kaiserslautern, Germany

†stenger@wiwi.uni-frankfurt.de, Chair of IT-based Logistics, Institute of Information Systems, Goethe University, Frankfurt am Main, Germany
min \sum_{i,j \in V', i \neq j} d_{ij}x_{ij} \quad (1)

\sum_{j \in V', i \neq j} x_{ij} = 1 \quad \forall i \in I \quad (2)

\sum_{j \in V', i \neq j} x_{ij} \leq 1 \quad \forall i \in F' \quad (3)

\sum_{i \in V', i \neq j} x_{ji} - \sum_{i \in V', i \neq j} x_{ij} = 0 \quad \forall j \in V' \quad (4)

\sum_{j \in V' \{v_0\}} x_{0j} \leq m \quad \forall j \in V' \quad (5)

\sum_{j \in V' \{v_0\}} x_{j0} \leq m \quad \forall j \in V' \quad (6)

0 \leq \tau_i + (t_{ij} + p_i)x_{ij} - T_{max}(1 - x_{ij}) \leq \tau_j \quad \forall i \in V', \forall j \in V' \{v_0\} \text{ and } i \neq j \quad (7)

\tau_j + t_{j0} + p_j \leq T_{max} \quad \forall j \in V' \{v_0\} \quad (8)

0 \leq \tau_0 \leq T_{max} \quad (9)

0 \leq y_j \leq y_i - (r \cdot d_{ij})x_{ij} + Q(1 - x_{ij}) \quad \forall j \in I, \forall i \in V' \text{ and } i \neq j \quad (10)

0 \leq y_i - (r \cdot d_{ij})x_{ij} \quad \forall j \in F_0, \forall i \in V' \text{ and } i \neq j \quad (11)

y_j = Q \quad \forall j \in F_0 \quad (12)

x_{ij} \in \{0, 1\} \quad \forall i, j \quad (13)

Note that the G-VRP benchmark instance files proposed in Erdogan and Miller-Hooks (2012) contain coordinates for each vertex \( v \) given in longitude and latitude. They have to be converted to distances between vertices \( v \) and \( w \) by applying the haversine formula using an earth radius of 4182.449. Further note that some customers contained in the test instances are infeasible, i.e., they cannot be served under the given restrictions. These customers have to be identified and removed in a preprocessing step.

B Design of EVRPTW Benchmark Instances

This section gives a complete, self-contained description of the generated E-VRPTW instances. We create two sets of benchmark instances for the E-VRPTW. A set of 56 large instances, each with 100 customers and 21 recharging stations, and a set of 36 small instances with 5, 10 and 15 customer per instance. All instances are created based on the benchmark instances for the VRPTW proposed by Solomon (1987). These instances are divided into 3 classes depending on the geographical distribution of the customer locations: Random customer distribution (R),
clustered customer distribution (C) and a mixture of both (RC). Groups R1, C1 and RC1 have a short scheduling horizon, meaning that generally more vehicles are required to serve all customers than in R2, C2 and RC2, which have a long scheduling horizon. The instances within a group differ in terms of time window density and time window width as shown in Table 1.

<table>
<thead>
<tr>
<th>#</th>
<th>C1 TWD</th>
<th>C1 TWW</th>
<th>C2 TWD</th>
<th>C2 TWW</th>
<th>R1 TWD</th>
<th>R1 TWW</th>
<th>R2 TWD</th>
<th>R2 TWW</th>
<th>RC1 TWD</th>
<th>RC1 TWW</th>
<th>RC2 TWD</th>
<th>RC2 TWW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>60.76</td>
<td>100%</td>
<td>160.00</td>
<td>100%</td>
<td>10.00</td>
<td>100%</td>
<td>115.96</td>
<td>100%</td>
<td>30.00</td>
<td>100%</td>
<td>120.00</td>
</tr>
<tr>
<td>2</td>
<td>75%</td>
<td>61.27</td>
<td>75%</td>
<td>160.00</td>
<td>75%</td>
<td>10.00</td>
<td>75%</td>
<td>111.80</td>
<td>75%</td>
<td>30.00</td>
<td>75%</td>
<td>120.00</td>
</tr>
<tr>
<td>3</td>
<td>50%</td>
<td>59.90</td>
<td>50%</td>
<td>160.00</td>
<td>50%</td>
<td>10.00</td>
<td>50%</td>
<td>117.34</td>
<td>50%</td>
<td>30.00</td>
<td>50%</td>
<td>120.00</td>
</tr>
<tr>
<td>4</td>
<td>25%</td>
<td>60.64</td>
<td>25%</td>
<td>160.00</td>
<td>25%</td>
<td>10.00</td>
<td>25%</td>
<td>120.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100%</td>
<td>121.61</td>
<td>100%</td>
<td>320.00</td>
<td>100%</td>
<td>30.00</td>
<td>100%</td>
<td>240.00</td>
<td>100%</td>
<td>60.00</td>
<td>100%</td>
<td>240.00</td>
</tr>
<tr>
<td>6</td>
<td>100%</td>
<td>156.15</td>
<td>100%</td>
<td>486.64</td>
<td>75%</td>
<td>30.00</td>
<td>75%</td>
<td>240.00</td>
<td>100%</td>
<td>88.21</td>
<td>100%</td>
<td>349.50</td>
</tr>
<tr>
<td>7</td>
<td>100%</td>
<td>180.00</td>
<td>100%</td>
<td>612.32</td>
<td>50%</td>
<td>30.00</td>
<td>50%</td>
<td>240.00</td>
<td>100%</td>
<td>112.33</td>
<td>100%</td>
<td>471.94</td>
</tr>
<tr>
<td>8</td>
<td>100%</td>
<td>243.28</td>
<td>100%</td>
<td>640.00</td>
<td>25%</td>
<td>30.00</td>
<td>25%</td>
<td>240.00</td>
<td>100%</td>
<td>349.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>100%</td>
<td>360.00</td>
<td>100%</td>
<td>58.89</td>
<td>100%</td>
<td>349.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100%</td>
<td>86.50</td>
<td>100%</td>
<td>363.27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>100%</td>
<td>93.10</td>
<td>100%</td>
<td>471.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>100%</td>
<td>117.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Time window density (TWD) and average time window width (TWW) for Solomon instances

To generate E-VRPTW instances based on VRPTW instances, we have to 1) position the recharging stations, 2) set battery capacities, recharging and consumption rates and 3) adjust time windows in order to generate feasible instances.

B.1 Location of Recharging Stations

We locate one recharging station at the depot because a recharging possibility at the depot seems to be a reasonable claim. The location of the remaining 20 stations is determined in a random manner. However, we limit the possible locations in order to generate feasible and meaningful instances, i.e., every customer can be reached from the depot using at most two different recharging stations. To this end, the recharging stations are placed within an area defined by three circles surrounding the depot. The circles are divided into equally spaced segments whose size depends on the number of stations to place, as illustrated in Figure 1. Two recharging stations are then randomly positioned into each segment, one between the inner and middle circle and the other between the middle and outer circle.

B.2 Battery Capacity, Recharging and Consumption Rate

The battery capacity is set to the maximum of the following two values: 1) the charge needed to travel 60% of the average route length of the best known solution to the corresponding VRPTW instance and 2) twice the amount of battery charge required to travel the longest arc between a customer and a station. This procedure ensures that instances with geographically disperse and remote customers stay feasible. Furthermore, we thus guarantee that recharging stations have to be used. For the sake of simplicity, we set the consumption rate to 1.0. The recharging rate
is set so that a complete recharge requires three times the average customer service time of the respective instance.

B.3 Adjustment of Time Windows

The detours for visits to recharging stations and the recharging times incurred make it impossible to comply with the customer time windows given in the original Solomon instances, i.e., some instances become infeasible because no possibility exists to reach certain customers within their original time window. Consequently, we have to generate new time windows to obtain feasible instances. The procedure used is very close to the original one described in Solomon (1987).

First, we determine for each customer the feasible time window range, i.e., the earliest time at which the customer is reachable from the depot (with potentially necessary station visits) and the latest time at which the customer must be left so that the depot can be reached in time. For instance sets R and RC, the time window center is randomly drawn from the determined interval. Like in Solomon (1987), the time window centers for set C are set corresponding to real arrival times, which we obtained by solving the instances without time windows using our VNS/TS.

Next, the time window widths are chosen according to the respective original instances. If a thus generated customer time window is outside the feasible range, the violating range is cut and the time window is extended to the other side of the interval.

B.4 Generation of Small-Sized Instances

To generate the set of small instances, we start with the 56 large instances described above. For each of the three sizes (5, 10, 15 customers), we randomly draw the respective number of customers from the large instances, thus generating 168 instances. The created instances are then solved with our VNS/TS heuristic and the solutions are inspected. For each problem
group and instance size, we select the two instances whose solution uses the highest number of recharging stations. In this way, we create $6 \cdot 3 \cdot 2 = 36$ small test instances, which are denoted by the identifier of the underlying Solomon instance followed by the number of customers in the instance, e.g., R108-5.

**References**
