Recharge at lunch, an alternative to handle the range issues of electric vehicles

David L. Cortés-Murcia, H. Murat Afsar, Caroline Prodhon
ICD-LOSI, Université de Technologie de Troyes, UMR-STMR CNRS 6281, 12 Marie Currie, CS42060, 10004 Troyes CEDEX, France.
{david.cortes_murcia,murat.afsar,caroline.prodhon}@utt.fr

Keywords: Operational research, exact method, branch-and-price, electric vehicles

1 Introduction

In order to reduce the negative environmental impacts and due to the cutting-edge technology, the governmental subsidies and new supportive policies, electric vehicles (EVs) are considered a suitable option for last mile operations [3][5]. However, according to [4], in France and UK the urban freight transport and service operators report some technical and operational obstacles that they have identified since they adopted electric vehicles. Two of those main obstacles are related to the limited range and the risk of queueing at recharging stations (RSs).

In this paper, we propose a novel variant of a routing problem where it is possible to recharge the EVs during the lunch time. This option was mentioned by two operators in [4] results, because lunch time is considered a mandatory stop. In this variant we assume that it is possible to have an agreement with a set of restaurants which provides a RS and a reservation for recharging during lunch. We aim to benefit from idle time caused for the lunch time and the priority that can be obtained from restaurants for recharging activities. A fixed cost is associated to visit each restaurant. It represents the cost established for restaurants to provide food and recharging service. Likewise, an estimated profit for visiting customers is defined. The objective is to maximize the operational benefit. Thus, the proposed problem generalizes the capacitated profitable tour problem (CPTP), where decision related to visit or not customers and restaurants must be made.

To solve the problem, we propose a Branch-and-Price algorithm and we test it on a set of instances adapted from instances proposed by [2]. Finally, computational tests to prove the efficiency of the algorithm are presented.

2 Problem statement and solution method

The problem is defined on a graph \( G = (V', A) \) with a set of vertices \( V' = \{V \cup F'\} \) and a set of arcs given by \( A = \{(i, j)|i, j \in V', i \neq j\} \). Let be \( V \) a set composed of customers and depot and \( F \) the set of restaurants who provides recharging stations. A homogeneous EV fleet of size \( U \) is available at the depot. For each visit to a restaurant, the lunch time is denoted as \( S \) and it is considered constant. Also, we assume that during that time the EV is charged to its maximum battery level \( B \). Energy consumption through an arc \( (i, j) \) is described as a linear relation between distance \( d_{ij} \) and the consumption rate \( c_r \).

Each customer \( i \in V \) has associated a positive score \( p_i \). Restaurants \( j \in F \) have a fix cost \( C_j \) associated to food and recharge service. Likewise, each restaurant has a hard time window \([e_j, l_j]\). It represents the lunch time established by the restaurant schedule or by the company policies. Lastly, a maximum tour duration is represented by \( T_{max} \).

The problem seeks to find at most \( U \) routes departing from the depot visiting a subset of customers. Each route includes one stop to visit a restaurant during its lunch time. The
objective function aims to maximize the total operational income computed as the sum of the profits minus the operational cost associated with the total distance and minus total cost associated with the restaurant service.

As a solution method we design a Branch-and-Price algorithm (B&P). Thus the problem is formulated as a set-packing model. From the set-packing model the dual is obtained, and the subproblem to be solved in the column generation phase is identified. In this case, it corresponds to an elementary longest path problem with resource constraints. A solution procedure based on a labeling algorithm is implemented to solve it. Finally, the branching strategies are adapted from those propose by [1], to be able to manage visits to restaurants.

3 Computational results and conclusions

A set of instances of this problem is created based on the set of instances proposed by [2]. The B&P is implemented in C++ and linear relaxations are solved by CPLEX 12.8. The algorithm is executed on a machine with Intel(R) Core(TM) i5-5300U CPU @2.30GHz and 8 GB RAM. The execution time is limited up to 2 hours.

<table>
<thead>
<tr>
<th>Set</th>
<th>n</th>
<th>rs</th>
<th>B\text{small}</th>
<th>B\text{medium}</th>
<th>B\text{large}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nbOp/nbInst</td>
<td>CPU(s)</td>
<td>nbOp/nbInst</td>
<td>CPU(s)</td>
<td>nbOp/nbInst</td>
</tr>
<tr>
<td>Set-1</td>
<td>30</td>
<td>4</td>
<td>54/54</td>
<td>0.04</td>
<td>54/54</td>
</tr>
<tr>
<td>Set-2</td>
<td>19</td>
<td>3</td>
<td>33/33</td>
<td>0.03</td>
<td>33/33</td>
</tr>
<tr>
<td>Set-3</td>
<td>31</td>
<td>4</td>
<td>60/60</td>
<td>14.22</td>
<td>55/60</td>
</tr>
<tr>
<td>Set-4</td>
<td>98</td>
<td>12</td>
<td>36/60</td>
<td>284.2</td>
<td>24/60</td>
</tr>
<tr>
<td>Set-5</td>
<td>64</td>
<td>8</td>
<td>65/78</td>
<td>141.93</td>
<td>51/78</td>
</tr>
<tr>
<td>Set-6</td>
<td>62</td>
<td>8</td>
<td>39/42</td>
<td>121.07</td>
<td>34/42</td>
</tr>
<tr>
<td>Set-7</td>
<td>100</td>
<td>13</td>
<td>38/60</td>
<td>110.87</td>
<td>31/60</td>
</tr>
</tbody>
</table>

Total 325/387 282/387 282/387

TAB. 1 – Computational results with different B values

In Table 1, we present the resume of the preliminary results. n is the number of available customers, rs is the number of available restaurants, nbOp/nbInst is number of optimal solutions reached by the algorithm over the set size, and CPU is the average computational time (in seconds) for those instances solved to optimality.

Preliminary results show that the proposed algorithm is capable to solve some instances with 100 customers and 13 restaurants, in less than 3 minutes on average. Those instances are commonly considered as big size instances for this type of problems.

Références