Heuristic algorithm based on columns generation for the Integrated Problem of Vehicle Routing and Container Loading

Diego Alejandro Acosta Rodríguez
David Álvarez Martínez

Department of Industrial Engineering
Universidad de los Andes
Bogotá, Colombia.

Abstract

This paper proposes a heuristic algorithm with a column generation structure, where the master problem is responsible for managing the selection of best set routes, while the slave problem is responsible for solving a shorter restricted route problem (CSP, Constrained Shortest Path) for the generation of columns (feasible routes). The CSP is not necessarily resolved to optimality. In addition to this, a GRASP algorithm (Greedy Randomized Adaptive Search Procedure) is used to verify packing constraints. The master problem begins with a set of feasible routes found through a multi-start randomized constructive heuristic (MSRCA, Multi-Start Randomized Constructive Algorithm) for the multi-container loading problem (3D-BPP, Three-dimensional Bin Packing Problem). The MSRCA consists in finding a group of valid routes thinking only in the best packing of the costumers (Packing First-Route Second). To validate the performance of the optimization methodology proposed here, a benchmark was made with the best solutions reported in the literature using the classic test sets. The results allow to conclude that the slave problem is too complex and computationally expensive to solve it through a MIP, as future proposals it is proposed the use of a labeling algorithm that allows finding a fast and good quality solution.

1. Introduction

The integrated problem of vehicle routing and packing of goods in three dimensions that will be dealt in this document, is categorized as a NP-Hard problem because it is a generalization of the problem of routes or the problem of packing [8]. A first approach to this problem is to consider only two dimensions at the time of loading the vehicle, this because in many applications of real-life products or goods are stowed, in the process of construction of pallets is set a maximum height and there is no possibility to stack one stowage on top of the other. On the other hand, in many logistic contexts the boxes are not stowed, so at the time of loading the vehicles it is necessary to consider solving the packing problem considering three dimensions.

The present project presents a heuristic solution for vehicle routing and merchandise packing (3L-CVRP). First, an approximation of the CVRP model will be made by generation of columns (CG), in which a master and an auxiliary problem will be obtained, the auxiliary problem will be a CSP that will yield feasible routes, with reduced costs of less than zero, to the master problem. On the other hand, the CG will use the GRASP model [1] to verify the feasibility of packing the route that
generated the CSP, this algorithm will give a positive response in which the column rises to the master problem or a negative, where it will generate a cut in the auxiliary problem and the problem will be re-optimized until a viable solution is found. In addition, a randomized multi-start construct algorithm is used to initialize the master problem, in which feasible routes will be obtained. Finally, the computational results and the respective conclusions of the model will be shown.

The document is organized as follows, in section 2 the state of the art is discussed, in section 3 the CVRP model is presented, the approximation of the generation of columns is in section 4, in section 5 is the GRASP model, the MSRCA is in section 6, in section 7 are the computational results and finally the conclusions are in section 8.

2. State of the art

Currently, there are a lot of proposed methodologies that address this same problem. Especially in the research groups COPA and PYLO of the Department of Industrial Engineering of the Universidad de los Andes there are two publications. The first is a hybrid algorithm based on the classic formulation of trained vehicle routing and a metaheuristic load-packing algorithm in three-dimensional form (3D-SLOPP, Three-Dimensional Single Large Object Packing Problem). By combining both, a series of cuts is made in the routes by the restrictions of subtours, capacity and packing. This is known as "A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3L-CVRP)" [5]. The second paper presents a metaheuristic algorithm known as "A hybrid metaheuristic approach for the Capacitated Vehicle Routing Problem with Container Loading Constraints" [6], in which the routing obtained by the Clarke & Wright algorithm is used, modified to guarantee the packing of the load by means of a GRASP algorithm. Subsequently, a granular search algorithm Tabu improves the previously obtained solution, while the GRASP validates the distribution of the load, finally, a solution of the problem is obtained. These two works are developed from different approaches, but both contain the heuristic aspect within their development, which generates that the solution is approximate.

The work "A tabu search algorithm for a routing and container loading problem" [9], contains a solution to the problem of 3L-CVRP using a Taboo search, and for each solution of the obtained routing this search solves the packing of the boxes, the route regardless of whether the load exceeds the length of the truck will be taken into account with a penalty. Another Taboo search applied to packaging routing is the one proposed in "A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem" [14], where they propose a Taboo search of its authorship in two dimensions and then approximate the three-dimensional solution in "Metaheuristics for vehicle routing problems with three-dimensional loading constraints" [7]. For the project carried out in "A hybrid algorithm for the capacitated vehicle routing problem with threedimensional loading constraints" [2] to solve the routing a Tabu search is proposed, later a tree algorithm develops the packing of the ordered routes by costs.

Additionally, there are works with heuristics as seen in "Packing first, routing second - a heuristic for the vehicle routing and loading problem" [3], where an optimal solution for the packing is
proposed first and then the routing is solved. On the other hand, there are the heuristic algorithms in the auxiliary problem, presented in "A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints" [11] where the heuristic loading algorithm is constructive and in "A hybrid approach for the vehicle routing problem with three-dimensional loading constraints" [13] the load problem is solved by a Tabu search.

The authors of "Exploring a Column Generation Approach for a Routing Problem with Sequential Packing Constraints" [12], propose an approximation by generation of columns for the routing and packing of goods in 2 dimensions, using cuts when the capacity of the vehicle is exceeded. While in "A column generation based heuristic for the capacitated vehicle routing problem with three-dimensional loading constraints" [10], the authors propose a solution based on the column generation heuristic technique, where the auxiliary problem solves a CSP through of a heuristic algorithm. In addition to this, vehicle loading restrictions are verified by a set of packaging heuristics.

Finally, the COPA research group at the Universidad de los Andes developed the "Pulse Framework for Hard Shortest Path Problems" method [4], this algorithm was implemented for CSP problems. The pulse as the authors call it, generates pruning by dominance, feasibility and by dimensions. As the authors note in this article this algorithm may take high computational times, this makes it not a good candidate for the work presented here, because a large number of CSP problems must be solved and the solution obtained can reach be discarded

3. Description of the capacitated routing problem

For the formulation of the capacitated routing problem (CVRP), we first define a set of homogeneous vehicles (V) and one of customers (C), where the depot is node 0. The sequence in which a group of customers is visited for each vehicle will be decided, in other words, \( x_{ijr} \) is defined as a binary variable that takes the value of 1 if the vehicle \( r \in V \) uses the arc linking the client \( i \in C \) with the client \( j \in C \) (11). Additionally, the truck has given dimensions \( (VolC) \) that can not be exceeded by the volume of the route, as seen in restriction 7, where each client has an associated volume \( v_i \). Similar to the volume, we have the load weight of each client \( p_i \), in the same way we can not exceed the weight \( (pesC) \) corresponding to the vehicles (8).

Due to the limited number of trucks that can be had, the number of routes cannot exceed the number of vehicles \( k \) (10). On the other hand, each client must be visited strictly once (2 and 3). Restrictions 4, 5 and 6 represent the balance equations, each route starts and ends in the depot (4) and (5), and everything that enters an \( i \in C \) node must exit (6). To avoid the subtours is restriction 9, where iteratively cuts will be added to restrict the number of clients of the subtour and thus prohibit the creation of them. Finally, the objective of the problem is to minimize the total distance of all routes (1).

\[
min \sum_{i \in C} \sum_{j \in C} \sum_{r \in R} x_{ijr} d_{ij} \quad s. a.
\]
(2) \[ \sum \sum_{r \in R, j \in C} x_{ijr} = 1 \quad \forall i \in C \]

(3) \[ \sum \sum_{r \in R, i \in C} x_{ijr} = 1 \quad \forall i \in C \]

(4) \[ \sum_{j \in C} x_{0jr} = 1 \quad \forall r \in V \]

(5) \[ \sum_{i \in C} x_{i0r} = 1 \quad \forall r \in V \]

(6) \[ \sum_{i \in C} x_{ihr} - \sum_{j \in C} x_{hjr} = 0 \quad \forall r \in V, \forall h \in C \]

(7) \[ \sum \sum_{i \in C, j \in C} x_{ijr} v_i p_i \leq VolC \quad \forall r \in V \]

(8) \[ \sum \sum_{i \in C, j \in C} x_{ijr} p_i \leq pesC \quad \forall r \in V \]

(9) \[ \sum \sum_{i \in C, j \in C} x_{ijr} \leq (Number Of Customers - 1) \quad \forall r \in V \]

(10) \[ \sum \sum_{j \in C, r \in V} x_{0jr} \leq k \]

(11) \[ x_{ijr} \in \{0,1\} \quad \forall i, j \in C, \forall r \in V \]

### 4. Approximation of the problem by columns generation (AHBGC)

The main contribution of this work is a heuristic algorithm based on columns generation for integrated routing and packing problem. It consists of dividing the problem into a master problem and an auxiliary problem. In addition to this, a communication is established where the master problem sends the values of their dual variables, while the auxiliary feedback the master problem to transmit their solution which is incorporated as a new column. The master problem is responsible for managing the sequences of the vehicles, to obtain \( k \) optimum routes that minimize the total distance and to send the value of its variables to the auxiliary problem. On the other hand, the auxiliary problem is responsible for generating feasible routes in terms of weight, volume and the dimensional restrictions of the items (the latter will be reviewed by a packing GRASP algorithm). The result of the auxiliary problem is transmitted to the master problem, so that it incorporates it in its model. A diagram of the approach proposed here is illustrated in Figure 1.
a. Master Problem (MP)

As for the master problem, it is going to declare a set of routes (R) and one of clients (C), where the depot is node 0. By the binary variable $y_r$ (15), defined as 1 if the route $r \in R$ or 0 of otherwise, the MP will ensure that each client is visited once (13), using the $a_{ir}$ parameter. This parameter will depend on the number of routes with reduced cost less than zero that are obtained from the auxiliary problem and will be defined mathematically later. In addition, the MP must ensure that the number of optimal routes is less than or equal to the number of vehicles $k$ (14). Finally, the objective function is to minimize the distance of the routes (12).

$$\begin{align*}
(12) & \quad \min \sum_{r \in R} c_r y_r \quad s. a. \\
(13) & \quad \sum_{r \in R} a_{ir} y_r \geq 1 \quad \forall i \in C, \pi \text{ dual variables} \\
(14) & \quad \sum_{r \in R} y_r \leq k \quad , 1 \text{ dual variable (}\sigma\text{)} \\
(15) & \quad y_r \geq 0 \quad \forall r \in R
\end{align*}$$

Once the master problem is solved, it will generate $\pi + 1$ dual variables ($W_i^r$) that will be used by the auxiliary problem in the objective function.
b. Auxiliary problem (AP)

For the auxiliary problem, a CSP is made, where the objective is to minimize the reduced cost of the original problem (15), that is, we want to achieve as many routes as possible until we find one with a value greater than or equal to $\sigma$ in the value of the function objective, this by means of a binary variable $x_{ij}$, which is defined as a binary variable that takes the value of 1 if the arc linking the client $i \in C$ with the client $j \in C$ is activated. Also, for this problem the remaining restrictions that were not used in the master problem are maintained. The restriction (16) and (17) keep to each client must be visited strictly once. The restrictions (18) and (19) ensure that the route starts and ends in the hold, while the restriction (20) makes that if an arc is activated ($x_{ij} = 1$) towards a client $j$ must also come out of the node ($x_{ji} = 1$). Additionally, the volume and weight of the truck cannot be exceeded, this is achieved through restrictions 21 and 22 respectively. To keep a route without subtours, an iterative system of cuts is implemented, where the clients that generate subtours and by means of restriction 23 the number of active nodes of that route will be restricted. In this way, the model will be reoptimized until there are no subtours in the solution. Finally, the nature of the variables (24).

\[
\begin{align*}
(15) \quad & \min \sum_{i \in C} \sum_{j \in C} x_{ij} \left( d_{ij} - \pi_t^i \right) \quad s. a. \\
(16) \quad & \sum_{j \in C} x_{ij} = 1 \quad \forall i \in C \\
(17) \quad & \sum_{i \in C} x_{ij} = 1 \quad \forall i \in C \\
(18) \quad & \sum_{j \in C} x_{0j} = 1 \\
(19) \quad & \sum_{i \in C} x_{i0} = 1 \\
(20) \quad & \sum_{i \in C} x_{ih} - \sum_{j \in C} x_{hj} = 0 \quad \forall h \in C \\
(21) \quad & \sum_{i \in C} \sum_{j \in C} x_{ij} v_l \leq VolC \\
(22) \quad & \sum_{i \in C} \sum_{j \in C} x_{ij} p_l \leq pesc \\
(23) \quad & \sum_{i \in C} \sum_{j \in C} x_{ij} < (Number\ of\ Customers - 1) \\
(24) \quad & x_{ij} \in \{0,1\} \quad \forall i, j \in C
\end{align*}
\]
Once the shortest route problem is solved, the parameter $a_{ir}$ (25) for the master problem is obtained.

\begin{equation}
25 \quad a_{ir} = \sum_{j \in C} x_{ij} \quad \forall r \in R \text{ generated}
\end{equation}

5. Packing Validator

In this work, the GRASP algorithm proposed in [1] was used as a validation routine for packing constraints. This algorithm is developed to solve the problem of three-dimensional packing considering restrictions of multiple destinations. In this way, when obtaining a solution of the auxiliary problem, the GRASP algorithm is called to validate if the column obtained meets the packing constraints. If it is a validated solution, it will become a new column of the master problem. Otherwise, this column is discarded and the auxiliary problem must be solved again until a valid column is obtained.

The packaging GRASP algorithm consists of two phases to verify the viability of the route. The first phase is construction and the second is an improvement phase. In the construction phase, a solution is created by adding the items step by step. In the second phase, the algorithm performs improvement movements that consist of emptying and filling again.

I. In the construction of the model a random strategy based on maximizing the use of the spaces or the box that best fits the available space is used, this process consists of 5 steps. In step zero, a list of empty three-dimensional spaces is created in the form of a parallelepiped. In the next step, the space with the greatest possible capacity is chosen from the list. In step two, the client is chosen and the items that will be packed depending on the chosen parallelepiped and two criteria, by volume which is the item that increases the volume occupied or the items that best fit according to their dimensions. In the third step, the list of spaces is updated generating new parallelepipeds for the current client, iteratively every time a new item is added. Finally, the list of parallelepipeds of remaining spaces for the next client is updated.

II. The randomization will allow to generate combinations of packaging with the items currently packed, and thus be able to occupy as much space as possible. This is achieved by creating combinations of packaging of the boxes, then randomly and based on the dimensions of the boxes and possible orientations to be packaged.

III. The improvement of the built solution will allow mobilizing and compressing the load to increase the occupied space. First, by means of a constructive deterministic method, where the boxes that were removed plus the ones that were not packed are repacked, based on the criterion of "best item that fits". The second part of the improvement is to perform the aforementioned method to a partial solution, the problem is considered improved when the occupied space decreases.
After the GRASP packing solution, it will return the information if the generated column can be packed in the truck or not. In the case that, if the route can be packed, it is sent to the master problem and the iteration of the generation of columns will follow, whereas, if the column is not feasible by packaging, the AP will be re-optimized by adding a cut that prohibits this obtained column (in equation 26 the cut is illustrated). This process is repeated until a new solution found is accepted by the packaging algorithm.

\[(26) \sum_{i \in C} \sum_{j \in C} x_{ij} < NumberOfCustomers\]

6. Multi-Start Randomized Constructive Algorithm (MSRCA)

To initialize the master problem, a multi-start randomized constructive algorithm (MSRCA) was used, consisting of 4 steps:

I. The list of vehicles and customers is initialized.
II. A vehicle (Vs) is selected depending on one of the following two criteria, with a probability of 50% each:
   i. Vehicle with less remaining volume (0% -50%)
   ii. Vehicle with less remaining weight (51% -100%)

The vehicles are ordered according to the chosen criterion and by equation (27) a pseudo-random probability is assigned to each vehicle, finally one is selected randomly.

III. To choose the customer is done in a similar way to the vehicles, one is chosen (Cs) based on 2 criteria with a probability of 50% each:
   i. Client with the highest volume (0% -50%)
   ii. Client with better weight adjustment (51% -100%)

Clients are ordered that can be packed according to the chosen criterion, if there are clients in the list by means of equation (27) a pseudo-random probability is assigned to each client, by the last one is selected in a random way. If there are no customers on the list, the vehicle is closed and you return to step 2.

IV. The list of customers and remaining vehicles is updated, the Vs route is updated with the Cs. In addition, if the available volume of the Vs is less than the minimum of the volumes of the remaining customers or if the available weight of the Vs is less than the minimum weight of the remaining customers, the car is closed and it returns to step 2
   i. If there are no vehicles available, but if remaining customers are returned to step 1, in the opposite case where there are vehicles and customers remaining, return to step 2.
An example of the pseudo-random probability would be: if I have 3 vehicles (a, b and c) homogeneous in weight and volume, for an intermediate iteration the truck has 30 units of remaining volume, the truck b has 50 units of remaining volume and the c has 20 units of remaining volume. In step 2, with a probability of 0.45, the criterion of the vehicle with the least volume is chosen. Sorting the trucks you get c, a and b, using equation (27).

\[
Pseudo - \text{random prob.} = \frac{(\text{SizeList} - \text{PositionCandidate} + 1)}{0.5 \times (\text{SizeList} + 1)}
\]

\[
Pseudo - \text{random prob.}(\text{truck a}) = \frac{(3 - 2 + 1)}{0.5 \times (3 + 1)} = \frac{2}{6}
\]

\[
Pseudo - \text{random prob.}(\text{truck b}) = \frac{3 - 3 + 1}{0.5 \times (3 + 1)} = \frac{1}{6}
\]

\[
Pseudo - \text{random prob.}(\text{truck c}) = \frac{3 - 1 + 1}{0.5 \times (3 + 1)} = \frac{1}{2}
\]

Therefore, from 0% to 50% probability the truck is chosen c, from 51% to 83.3% the truck is chosen and from 83.4% to 100% the truck is chosen b. If a random number of 0.38 results, the vehicle c is chosen.

In this way, the MSRCA algorithm was used to obtain the routes that will initialize the \( a_{ir} \) parameter of the master problem.

7. Computational results

The implementation of the MSRCA algorithm was performed on a computer with the following specifications: Windows 7 Enterprise®, with an Intel® Core™ i7-4610M CPU @ 3.0 GHz and 16 GB of RAM. For the realization of the CVRP model by columns generation and the GRASP algorithm, was used on a computer with the following specifications: Lenovo Legion Y520 – Linux Ubuntu 16.04 LTS - Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 8 and 15.5 GB of RAM. To implement the algorithms was used the optimization IBM CPLEX Studio 12.6®×64-bit program, which is among the three best optimizers of linear programming at present, and C++™ programming language, this allowed the development of the heuristic algorithm code by its programming language to objects, in addition of it’s friendly handling with the user. The reactive GRASP algorithm proposed in [1] was used in this work both to verify the solutions obtained by the auxiliary problem as an element of the MSRCA algorithm. This algorithm requires parameter calibration because it depends on a maximum number of iterations. In this study the proposed value [6] was used for this parameter. On the other hand, given that the auxiliary problem is not necessarily resolved to optimality, it was necessary to calibrate the parameter of maximum
In the first instance, the MSRCA algorithm was tested in the instances of the literature. The results obtained are shown in Table 1. As can be seen in the first column, we have the 27 instances for which it was tested, followed by each of its number of customers, items and vehicles. Additionally, we have the number of starts that the algorithm had to perform to find a solution. Finally, in the
last column there is the duration time, in seconds, that the MSRCA used to find the solution of the routes for each instance, demonstrating its high effectiveness.

Table 2. Computational results of problem 3L-CVRP

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Items</th>
<th>Vehicles</th>
<th>BKS</th>
<th>Gendreau Solution</th>
<th>Tarantilis Solution</th>
<th>Bortfeldt Solution</th>
<th>GTS-Grasp Solution</th>
<th>AHBGC Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>32</td>
<td>4</td>
<td>300.70</td>
<td>316.32</td>
<td>1800</td>
<td>321.47</td>
<td>13</td>
<td>302.02</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
<td>5</td>
<td>334.96</td>
<td>350.58</td>
<td>1800</td>
<td>334.96</td>
<td>12</td>
<td>334.96</td>
</tr>
<tr>
<td>20</td>
<td>37</td>
<td>4</td>
<td>392.63</td>
<td>447.73</td>
<td>1800</td>
<td>430.95</td>
<td>54</td>
<td>392.63</td>
</tr>
<tr>
<td>20</td>
<td>36</td>
<td>6</td>
<td>430.89</td>
<td>448.48</td>
<td>1800</td>
<td>458.04</td>
<td>324</td>
<td>437.19</td>
</tr>
<tr>
<td>21</td>
<td>45</td>
<td>6</td>
<td>443.61</td>
<td>464.24</td>
<td>1800</td>
<td>465.79</td>
<td>100</td>
<td>443.61</td>
</tr>
<tr>
<td>21</td>
<td>40</td>
<td>6</td>
<td>498.16</td>
<td>504.46</td>
<td>1800</td>
<td>507.96</td>
<td>1212</td>
<td>498.16</td>
</tr>
<tr>
<td>22</td>
<td>46</td>
<td>6</td>
<td>769.68</td>
<td>831.66</td>
<td>1800</td>
<td>796.61</td>
<td>365</td>
<td>769.68</td>
</tr>
<tr>
<td>22</td>
<td>43</td>
<td>6</td>
<td>810.89</td>
<td>871.77</td>
<td>1800</td>
<td>880.93</td>
<td>230</td>
<td>810.89</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>8</td>
<td>630.13</td>
<td>666.10</td>
<td>1800</td>
<td>642.22</td>
<td>982</td>
<td>630.13</td>
</tr>
<tr>
<td>29</td>
<td>62</td>
<td>8</td>
<td>820.35</td>
<td>911.16</td>
<td>3600</td>
<td>884.74</td>
<td>1308</td>
<td>820.35</td>
</tr>
<tr>
<td>29</td>
<td>58</td>
<td>8</td>
<td>762.51</td>
<td>819.36</td>
<td>3600</td>
<td>873.43</td>
<td>523</td>
<td>803.61</td>
</tr>
<tr>
<td>30</td>
<td>63</td>
<td>9</td>
<td>614.59</td>
<td>651.58</td>
<td>3600</td>
<td>624.24</td>
<td>295</td>
<td>614.59</td>
</tr>
<tr>
<td>32</td>
<td>61</td>
<td>8</td>
<td>2645.95</td>
<td>2928.34</td>
<td>3600</td>
<td>2799.74</td>
<td>2193</td>
<td>2645.95</td>
</tr>
<tr>
<td>32</td>
<td>72</td>
<td>9</td>
<td>1368.42</td>
<td>1559.64</td>
<td>3600</td>
<td>1504.44</td>
<td>4581</td>
<td>1368.42</td>
</tr>
<tr>
<td>32</td>
<td>68</td>
<td>9</td>
<td>1341.14</td>
<td>1452.34</td>
<td>3600</td>
<td>1415.42</td>
<td>2528</td>
<td>1341.14</td>
</tr>
<tr>
<td>35</td>
<td>63</td>
<td>11</td>
<td>698.61</td>
<td>707.85</td>
<td>3600</td>
<td>698.61</td>
<td>4257</td>
<td>698.61</td>
</tr>
<tr>
<td>40</td>
<td>79</td>
<td>14</td>
<td>824.01</td>
<td>920.87</td>
<td>3600</td>
<td>872.79</td>
<td>2096</td>
<td>866.40</td>
</tr>
<tr>
<td>50</td>
<td>99</td>
<td>12</td>
<td>741.74</td>
<td>871.29</td>
<td>7200</td>
<td>818.68</td>
<td>2509</td>
<td>741.74</td>
</tr>
<tr>
<td>71</td>
<td>147</td>
<td>18</td>
<td>587.95</td>
<td>732.12</td>
<td>7200</td>
<td>641.57</td>
<td>1941</td>
<td>587.95</td>
</tr>
<tr>
<td>75</td>
<td>155</td>
<td>17</td>
<td>1090.22</td>
<td>1275.20</td>
<td>7200</td>
<td>1159.72</td>
<td>2823</td>
<td>1090.22</td>
</tr>
<tr>
<td>75</td>
<td>146</td>
<td>18</td>
<td>1147.80</td>
<td>1277.94</td>
<td>7200</td>
<td>1245.35</td>
<td>2686</td>
<td>1147.80</td>
</tr>
<tr>
<td>75</td>
<td>150</td>
<td>17</td>
<td>1130.54</td>
<td>1258.16</td>
<td>7200</td>
<td>1231.92</td>
<td>4659</td>
<td>1130.54</td>
</tr>
<tr>
<td>75</td>
<td>143</td>
<td>16</td>
<td>1116.13</td>
<td>1307.09</td>
<td>7200</td>
<td>1201.96</td>
<td>4854</td>
<td>1116.13</td>
</tr>
<tr>
<td>100</td>
<td>193</td>
<td>22</td>
<td>1407.36</td>
<td>1570.72</td>
<td>7200</td>
<td>1457.46</td>
<td>5726</td>
<td>1407.36</td>
</tr>
<tr>
<td>100</td>
<td>199</td>
<td>26</td>
<td>1600.35</td>
<td>1847.95</td>
<td>7200</td>
<td>1711.93</td>
<td>6283</td>
<td>1600.35</td>
</tr>
<tr>
<td>100</td>
<td>198</td>
<td>23</td>
<td>1529.86</td>
<td>1747.52</td>
<td>7200</td>
<td>1646.44</td>
<td>9916</td>
<td>1529.86</td>
</tr>
<tr>
<td>AVG TIME</td>
<td></td>
<td></td>
<td>4223</td>
<td>2421</td>
<td>225</td>
<td>1800</td>
<td>655</td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td></td>
<td>924.58</td>
<td>1028.48</td>
<td>985.67</td>
<td>928.09</td>
<td>966.68</td>
<td>1874.85</td>
</tr>
<tr>
<td>GAP RELATIVO %</td>
<td></td>
<td></td>
<td>82.3</td>
<td>90.2</td>
<td>102.0</td>
<td>93.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAP %</td>
<td></td>
<td></td>
<td>10.1</td>
<td>6.2</td>
<td>0.4</td>
<td>4.4</td>
<td>50.7</td>
<td></td>
</tr>
</tbody>
</table>

* BKS, Best Known Solution is calculated from the compilation of the different previous works that reached this solution in the least computation time.

† All computational times are reported in seconds.

To validate the efficiency of the AHBGC algorithm, a set of instances of the specialized literature was used, for this the quality of the obtained solution and the computational time necessary to reach it were taken into account. The results obtained are compared with the best solutions found for all the previous works for each of the instances. The results are shown in Table 2. As can be seen in the first column of the table, there are the number of clients for the 25 instances with which the work was tested. Followed by this, we have the best solution found (BKS) published by the different previous works in the literature. Then four representative works are shown, each
one with its respective total distance of the routes and the time used, in seconds, to find the solution. In the last column is the solution thrown by this project, it also has its respective total distance of the route and the time spent to find this answer. In addition, Table 2 has the information of the average distance of the routes and the average time spent by each author, finally we have the information of the GAP (%) with respect to the values of the BKS and the relative GAP (%) for each solution.

Analysis of results

For the instances used in this work, the MSRCA algorithm found for each of these always a feasible solution for the problem of 3D-BPP, where the most basic problem has 26 boxes and 15 clients, while the most complex problem has 199 boxes and 100 customers to first pack and then route. As can be seen from Table 1, the algorithm has a very competitive time due to its speed to find the solution, with an average of 5.2 seconds for the 27 instances.

Table 2 shows that the results obtained have only been tested in 26 instances of the literature, where the smallest problem has 15 clients and 4 vehicles, and the largest problem has 100 customers and 26 vehicles. The results show a stable behavior, although when increasing the number of clients of the problem there is an overflow of distance and time. Given that as the number of customers increases, the auxiliary problem presents great inconveniences due to its complexity. It is important to note that competitive results have been obtained in an instance with less than 50 clients, in addition to this, the average computation time is still very competitive in this area, although analyzing the total distances of the routes is still at a disadvantage. Regarding the percentage of the relative GAP of the average distance, it is obtained that the solution of this project is 100.6% far from the best solution presented by another author, being a large gap. On the other hand, the percentage of the GAP with respect to the BKS obtains a value of 50.4%, and although it is not the best solution if competitive results are presented and which will serve as a reference for future work.

8. Conclusions and future work

This document presents a heuristic solution based on generation of columns for the problem of vehicle routing and packaging of goods, which consisted of finding feasible routes in terms of volume, weight and load inside the truck, and finally minimize the distances of these routes. In addition to this, the formulation and the different algorithms used to carry out the project were presented. Also, the results were presented for different instances of the literature.

Regarding the behavior of the model, it can be said that, on a small scale the model presents competitive results, while on a large scale there is an overflow of time and distance, because the auxiliary problem is another NP-Hard problem. Finding optimal solutions for this model becomes very difficult. Therefore, the performance of the AHBGC algorithm depends on the structure of the AP model, which in turn is given by the number of clients to address the problem.
As future work we want to implement acceleration methods for the auxiliary problem. In addition, an algorithm will be developed that complements the columns that generate the column generation, since when a new column is added to the master problem it is not ensured that it is between the base, so when selecting a column with favorable reduced cost the algorithm will complement the path with other routes so that it enters the base. On the other hand, an unused column control will be taken to remove it from the route matrix and decrease the size of the problem.

References


