International Journal of Industrial Engineering Computations 13 (2022) 421–434

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A hybrid matheuristic approach for the vehicle routing problem with three-dimensional loading constraints

Diego Alejandro Acosta Rodríguez[a], David Álvarez Martínez[a] and John Willmer Escobar[b*]

[a]Department of Industrial Engineering, Universidad de Los Andes, Colombia
[b]Department of Accounting and Finance, Universidad del Valle, Colombia

C H R O N I C L E

Article history:
Received November 5 2021
Received in Revised Format December 21 2021
Accepted January 17 2022
Available online January, 17 2022

Keywords:
Column Generation
3L–CVRP
Heuristic
GRASP

A B S T R A C T

This paper proposes a matheuristic algorithm based on a column generation structure for the capacitated vehicle routing problem with three-dimensional loading constraints (3L–CVRP). In the column generation approach, the master problem is responsible for managing the selection of best-set routes. In contrast, the slave problem is responsible for solving a shorter restricted route problem (CSP, Constrained Shortest Path) for generating columns (feasible routes). The CSP is not necessarily solved to optimality. In addition, a greedy randomized adaptive search procedure (GRASP) algorithm is used to verify the packing constraints. The master problem begins with a set of feasible routes obtained through a multi-start randomized constructive algorithm (MSRCA) heuristic for the multi-container loading problem (3D–BPP, three-dimensional bin packing problem). The MSRCA consists of finding valid routes considering the customers' best packing (packing first-route second). The efficiency of the proposed approach has been validated by a set of benchmark instances from the literature. The results show the efficiency of the proposed approach and conclude that the slave problem is too complex and computationally expensive to solve through a MIP.

1. Introduction

The integrated vehicle routing and packing problem, considering three-dimensional loading constraints (3L–CVRP), is categorized as an NP-hard problem because it is a generalization of two well-known problems: the vehicle routing problem and the packing problem (Garey & Johnson, 1979). Commonly, routing and packing problems consider general two-dimensional space constraints because they could be applied for several real-life cases. In this case, the pallet construction is set with a fixed maximum height, and there is no possibility of stacking products. However, the packing problem in many logistic contexts is usually a three-dimensional case. An influential research field is the decision-making process for routing decisions considering packing aspects for a set of customers. Both decisions are classic, complex, well-known combinatorial problems. The combined problem of routing and packaging has increased the attention of industry and practitioners of applied science. The main aim of this paper is to merge the container loading problem (CLP) with the conventional vehicle routing problem (VRP) by using three-dimensional (3D) constraints. The CLP seeks to identify the optimal place for shipping boxes to a set of customers. The CLP's key goal is to optimize the used space and inventory distribution without relocation aspects when a customer is visited by a vehicle on a given route. In addition, the VRP aims to minimize the distance traveled by the vehicles (Bernal et al., 2018; Linfati & Escobar, 2018). The capacitated vehicle routing problem with three-dimensional loading constraints (3L–CVRP) occurs when the VRP and CLP are combined. The 3L–CVRP considers the combination of two well-known NP-hard problems: the capacitated vehicle routing problem (CVRP) and the three-dimensional loading problem (3L).

* Corresponding author
E-mail: john.wilmer.escobar@correounivalle.edu.co (J. W. Escobar)

2022 Growing Science Ltd.
doi: 10.5267/j.ijiec.2022.1.002

Let m_i be defined as the number of 3-dimensional items demanded by each customer i, and c_i be the total weight of the demand ordered by each customer $i\ \{i = 1,2,\dots,n\}$. Each item I_{il} has a width w_{il}, height h_{il}, and length l_{il}, where $\{l = 1,2,\dots,m\}$ indicates the different items ordered by a customer i. The 3L–CVRP considers a homogeneous fleet of v vehicles. Each vehicle's load surface has a width W, height H, length L, and limited capacity of weight C. Additionally, let $S(k) \subseteq \{i = 1,2,\dots,n\}$ be the set of customers visited by vehicle k. The 3L–CVRP considers the well-known constraints of the CVRP with practical 3-dimensional constraints. The total load of each $S(k)$ customer must fit into a container with $W \times H \times L$ dimensions. For the 3L–CVRP, the constraints associated with orientation, fragility, support area, and sequential loading must be satisfied. The orientation mentions that the item could be rotated without considering the vertical axis. The fragility constraints consider that each item I_{il} has the given fragility flag f_{il}. In particular, we consider that fragile goods must not be placed by any other object. The support area considerations require that each item must have a minimum percentage of the bottom area that is being supported. Sequential loading constraints establish a visit sequence of the customers and a sequence of loading and unloading of products. Indeed, this work considers all the packing constraints by characterizing them. Most of the previously published works for 3L–CVRP have eliminated some of the packing constraints. Concerning the vehicle capacity constraints, the previously published works do not consider that the type of product density of the boxes is assumed as 1, being the weight of each box equal to its volume.

We propose a novel heuristic approach for the 3L–CVRP based on Column Generation (CG). First, an approximation of the CVRP model is performed by CG with a master and slave problem. The slave problem is a Constrained Shortest Path (CSP) that yields feasible routes, with reduced costs of less than zero, to the master problem. On the other hand, the CG uses a GRASP approach to verify the feasibility of packing the routes generated by the CSP. The proposed algorithm gives positive or negative answers depending on whether the column rises to the master problem (positive) or if the approach generates a cut in the slave problem (negative). The problem is reoptimized until a feasible solution is obtained. A randomized multi-start constructive algorithm is employed to initialize the master problem to obtain feasible routes.

The paper is structured as follows: The literature associated with the 3L–CVRP is presented in Section 2. Section 3 presents a detailed description of the framework utilized by the proposed algorithm. A computational comparative study on a subset of benchmark instances from the literature is provided in Section 4. Section 5 contains concluding remarks and future research.

2. Literature review

Several researchers have investigated the 3L–CVRP because this problem has many practical applications. Exact algorithms for vehicle routing problems with loading constraints have been proposed by Junqueira et al. (2013), Hokama et al. (2016), Mahvash et al. (2017), Mak-Hau et al. (2018). Junqueira et al. (2013) propose an integer linear programming model for the 3L–CVRP. The proposed model considers constraints related to the cargo's vertical stability, multidrop situations, and load-bearing strength of the boxes (including fragility). Hokama et al. (2016) describe a branch-and-cut algorithm for the vehicle routing problem with unloading constraints. The authors consider the versions of the problem with two and tri-dimensional parallelepiped items. The proposed approach uses several techniques to prune the branch-and-cut enumeration tree. The presented algorithm uses several packing routines with different algorithmic approaches, such as branch-and-bound, constraint programming, and metaheuristics. Mahvash et al. (2017) present a CG technique-based heuristic to solve this problem. First, an elementary shortest path problem is solved to find routes with a negative reduced cost to generate new columns in the CG technique. Mak-Hau et al. (2018) present an exact integer linear programming model that serves two purposes: 1) providing exact solutions for problems of modest size as a basis for comparing the quality of heuristic solution methodologies and 2) for further exploration of various relaxations, stack generation, and decomposition strategies that are based on the ILP model.

Population-based methods for combined routing and packing problems have been introduced by Miao et al. (2012), Ruan et al. (2013), Lacomme et al. (2013), and Zhang et al. (2015), and recently by Vega-Mejía et al. (2020). Miao et al. (2012) solve the 3L–CVRP by a hybrid approach, which combines a genetic algorithm (GA) and Tabu Search (TS). Genetic Algorithm (GA) is developed for vehicle routing, and TS is developed for three-dimensional loading; these two algorithms are integrated for the combinatorial problem. Ruan et al. (2013) present a hybrid approach that combines honey bee matching optimization (HBMO) and six loading heuristics to solve the integrated problem. A new way to solve the packing subproblem of the 3L–CVRP was proposed by Lacomme et al. (2013). The proposed packing approach is included in a GRASP×ELS hybrid algorithm dedicated to the computation of VRP routes. A Local Search is defined on each search space. Zhang et al. (2015) introduce a new practical variant of the combined routing and loading problem, referred to as the capacitated vehicle routing problem, by minimizing fuel consumption under three-dimensional loading constraints (3L–FCVRP). A local evolutionary search (ELS) framework incorporating a recombination method is employed to explore the solution space. A new heuristic based on open space is applied to examine the feasibility of the solutions. Vega-Mejía et al. (2020) propose a hybrid heuristic method based on the GRASP metaheuristic and the Clarke and Wright Savings algorithm to solve a vehicle routing problem with several loading and new routing constraints. A TS strategy applied to combined packaging-routing problems was proposed by Tarantilis et al. (2009). A methodology that combines TS and guided local search (GLS) strategies is proposed. The loading characteristics are approached by employing a collection of packing heuristics. The authors also expand the work proposed for the two-dimensional load capacitated vehicle routing problem (2L-CVRP) in Fuellerer et al. (2010) to the 3L–CVRP. Bortfeldt et al. (2012) introduce an efficient hybrid algorithm including a TS algorithm for routing and a tree search algorithm for loading constraints. A local search (LS) algorithm for solving the three loading heterogeneous vehicle routing problem with time windows (3L–HFCVRPTW) was explored by Pace et al. (2015). This paper introduces the problem and develops a specialized procedure for loading goods. The authors apply simple local search procedures to the routing problem. Simulated annealing outperforms iterated local search, suggesting that the routing problem is multimodal and that operators shift deliveries between routes.

Bortfeldt et al. (2015) extend the vehicle routing problem with clustered backhauls (VRPCB) to an integrated routing and three-dimensional loading problem named VRPCB with 3D loading constraints (3L–VRPCB). The authors propose two hybrid algorithms for solving the 3L–VRPCB, consisting of routing and packing procedures. The routing procedures follow different metaheuristic strategies (large variable neighborhood search), and in both algorithms, a tree search heuristic is responsible for packing boxes. Männel & Bortfeldt (2016) consider the classical pickup and delivery problem (PDP) to be an integrated routing and three-dimensional loading problem, which is referred to as PDP with three-dimensional loading constraints (3L–PDP). The routing procedures modify a well-known large neighborhood search for the 1D-PDP. A tree search heuristic is responsible for packing boxes. Wei et al. (2018) study the well-known 2L–CVRP. A simulated annealing algorithm with a mechanism of repeated cooling and heating is proposed to solve the four versions of this problem, with or without the LIFO constraint and rotation of goods.

Tao & Wang (2010) propose a minor packing heuristic-based approach for solving the loading subproblem, which is iteratively invoked by a simple TS algorithm for the routing problem. Bortfeldt & Homberger (2013) consider the 3L–CVRP. A two-stage heuristic is presented following a "packing first, routing second" approach, i.e., packing goods and routing of vehicles is performed in two strictly separated stages. Indeed, an optimal solution for the packing is proposed, and then the routing problem is solved. Junqueira & Morabito (2015) present heuristic algorithms for 3L–CVRP for a real case situation. The objective is to identify minimum-cost delivery routes for a set of identical vehicles that, after departing from a depot, visit all customers only once and return to the depot. The proposed approaches are based on classical heuristics from both vehicle routing and container loading literature and two metaheuristic strategies and their use in more elaborate procedures. Escobar-Falcón et al. (2016) propose a hybrid algorithm based on the classic formulation of the vehicle routing problem and the load-packing algorithm in three-dimensional space (3D-SLOPP, three-dimensional single large object packing problem). Several cuts are performed in the routes by the constraints of sub tours, capacity, and packing. Escobar L.M. et al. (2015) introduce a hybrid metaheuristic approach for the 3L-CVRP. The proposed approach uses an initial solution obtained by a modified Clark & Wright algorithm considering a GRASP scheme's approach. A Granular Tabu Search (GTS) algorithm is then employed to improve the initial solution. The GRASP approach validates the packing constraints during the search process. These two works are developed from different approaches, but both contain the heuristic aspect of their development, which generates the approximate solution. Pinto et al. (2015) propose an approximation by generating columns for the routing and packing goods in two dimensions, using cuts when a vehicle's capacity is exceeded. Mahvash et al. (2017) propose a solution based on a column generation heuristic technique, where the slave problem solves a CSP through a heuristic algorithm. In addition, vehicle loading restrictions are verified by a set of packaging heuristics.

3. Description of the proposed approach

Since solution methods based on column generation for VRP problems are efficient, we propose a heuristic algorithm that combines exact methods and a GRASP procedure to solve the 3L–CVRP. In particular, we solve the CVRP problem using column generation while the packing constraints are satisfied by a GRASP algorithm.

3.1 Mathematical Formulation for the Capacitated Vehicle Routing Problem (CVRP)

For the formulation of the capacitated routing problem (CVRP), first, we define a set of homogeneous vehicles (V) and one of the customers (C), where the depot is node 0. The sequence in which a group of customers is visited for each vehicle is decided. Here, x_{ijr} is defined as a binary variable that takes the value of 1 if vehicle $r \epsilon V$ uses the arc linking customer $i \epsilon C$ with customer $j \epsilon C$ (Eq. (11)). Additionally, the truck has given dimensions ($VolC$) that cannot be exceeded by the route's volume, as seen in Constraint (7), where each customer has an associated volume v_i. Similar to the volume, we have the load weight of each customer p_i. Therefore, we cannot exceed the weight ($WeiC$) of vehicle (see Eq. (8)). On the other hand, each customer must be visited strictly once (Eq. (2) and Eq. (3)). Constraints (4) to (6) represent the balance equations. Each route starts and ends at depots (Eq. (4) and Eq. (5)), and everything that enters a node must exit (see Eq. (6)). The sub tour constraints are avoided by Eq. (9), where iterative cuts are added to restrict the number of customers of the sub tour and prohibit their creation. Due to the limited number of available trucks, the number of routes cannot exceed the *N* number of vehicles (see Eq. 10). The problem's objective is to minimize the total distance of all routes (Eq. (1)).

$$min \sum_{i \in C} \sum_{j \in C} \sum_{r \in R} x_{ijr} \, d_{ij} \qquad (1)$$

s.t:

$$\sum_{j \in C} \sum_{r \in R} x_{ijr} = 1 \qquad \forall i \in C \qquad (2)$$

$$\sum_{i \in C} \sum_{r \in R} x_{ijr} = 1 \qquad \forall j \in C \qquad (3)$$

$$\sum_{j \in C} x_{0jr} = 1 \qquad \forall r \in V \qquad (4)$$

$$\sum_{i \in C} x_{i0r} = 1 \qquad \forall r \in V \qquad (5)$$

$$\sum_{i \in C} x_{ihr} - \sum_{i \in C} x_{hjr} = 0 \qquad \forall r \in V;\ h \in C \qquad (6)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijr}\, v_i \leq VolC \qquad \forall r \in V \qquad (7)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijr}\, p_i \leq WeiC \qquad \forall r \in V \qquad (8)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijr} \leq |S| - 1 \qquad \forall\ |S| \geq 1. S \subseteq V \qquad (9)$$

$$\sum_{j \in C} \sum_{r \in V} x_{0jr} \leq N \qquad (10)$$

$$x_{ijr} \in \{0.1\} \qquad \forall r \in V; i.j\ \in C \qquad (11)$$

3.3 3L–CVRP solution by column generation (AHBGC)

One of this work's main contributions is a heuristic algorithm based on column generation for integrated routing and packing problems. The algorithm consists of splitting the problem into a master and a slave problem.

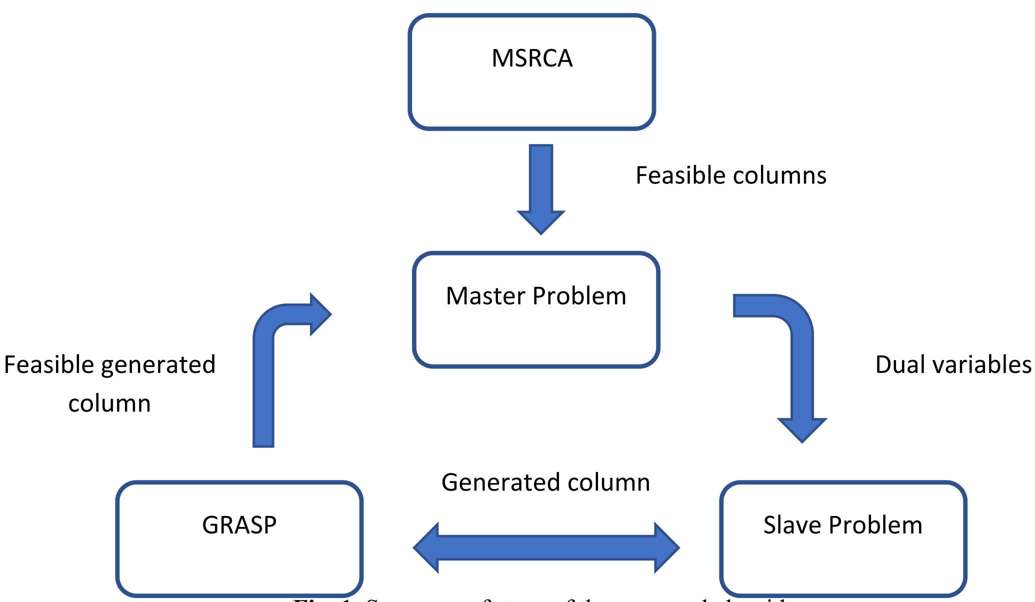

Fig. 1. Sequence of steps of the proposed algorithm

In addition, communication is established where the master problem sends the values of their dual variables. In contrast, slave feedback sends the master problem to transmit their solution, which is incorporated as a new column. The master problem is responsible for managing the vehicles' sequences to obtain optimum routes that minimize the total distance and send its variables to the slave problem. On the other hand, the slave problem is responsible for generating feasible routes in weight, volume, and dimensional constraints (by the packing GRASP algorithm detailed in this section). The results of the slave problem are transmitted to the master problem to be incorporated into the model. A diagram of the approach proposed here is illustrated in Fig. 1. A more detailed sequence of steps is presented in Algorithm 1, in it must be tunned two sets of parameters, the solver parameters that, in this case, are the default values for the CPLEX ® and the GRASP parameters that are detailed moreover. As input, the algorithm needs the information problem: the fleet of vehicles, the set of customers and its boxes, the size of the fleet, the distance matrix between depot and customers, and the weight and the volume limit of the vehicles. The column generation starts with a feasible set of routes obtained by a Multi start Randomized Constructive Algorithm (line 1) to create the master problem (line 2). Then, an iterative cycle is imposed (lines 3-23); the general idea on each iteration of this process is to solve the master problem to obtain a set of feasible routes that solve the problem (*incumbent*), and at the same time extract the dual variables ($\boldsymbol{W_i^t}$) and with the help of the slave problem propose new routes that end up updating the *incumbent* (line 6). Thus, the slave problem is generated (line 7), and internally another cycle is executed (lines 8-22); it can be found a new route or the end of the optimization process solving the slave problem (line 10). Whether a new route is found (lines 11-19), it must be checked for packing feasibility (using the GRASP Algorithm, line 12), every feasible route is sent to the master problem, breaking the internal loop, and continuing the outer cycle (lines 13-15), for unfeasible routes a cut to the slave problem is added (line 17) and continue iterating inside the inner cycle. When the solution of the slave problem is unfeasible, the complete process stops, returning the *incumbent* of the column generation algorithm (line 20).

Algorithm 1. **Column Generation Algorithm**

Parameters: *SolverP:* set of parameters required for the selected solver, *ParGRASP*: parameters required for the proposed GRASP;
Input: List *V: list of vehicles*, **List** *B*: list of boxes of the customers, *N*: number of available vehicles, d_{ij}: distance matrix, *WeiC*: weight limit of the vehicles, *VolC*: volume limit of the vehicles;
Output: Routes *Incumbent*: set of routes.

1: *Columns* ← *MSRCA*(*V*, *B*, *N*)
2: *MasterModel* ← *GenerateModel*(*Columns*, d_{ij}, *N*)
3: *FlagCycle* ← **true**
4: **While** *FlagCycle*
5:     *FlagCycle* ← **false**
6:     *Incumbent*, $\boldsymbol{W_i^t}$←*Solve*(*SolverP*; *MasterModel*)
7:     *SlaveModel* ← *GenerateModel*(W_i^t, d_{ij}, *WeiC*, *VolC*)
8:     *FlagColumn* ← **true**
9:     **While** *FlagColumn*
10:         *Route, Feasible* ←*Solve*(*SolverP*; *SlaveModel)*
11:         **if** *Feasible* **then**
12:             **if** *GRASPAlgorithm*(*ParGRASP; Route, B, V.dimensions)* **then**
13:                 *MasterModel* ← *AddColumn*(*Route*, *MasterModel*)
14:                 *FlagColumn* ← **false**
15:                 *FlagCycle* ← **true**
16:             **else**
17:                 *SlaveModel* ← *AddCut*(*Route*, *SlaveModel*)
18:             **end if**
19:     **else**
20:             **return** *Incumbent*
21:         **end if**
22:     **end while**
23: **end while**

3.2.1 Master Problem (MP)

For the master problem, it is essential to declare a set of routes (Rs) and one set of customers (C), where the depot node is 0. The binary variable y_r (see Eq. (15)) is defined as 1 if the route $r\epsilon Rs$ or 0 otherwise, and the MP ensures that each customer is visited once (Eq. (13)) using the a_{jr} parameter.

$$min \sum_{r\in R} c_r y_r \tag{12}$$

s.t:

$$\sum_{r\in R} a_{jr} y_r \geq 1 \qquad \forall j \in C.\ \pi_j\ dual\ variables \tag{13}$$

$$\sum_{r \in R} y_r \leq N \qquad \forall\ dual\ variable\ (\sigma) \qquad (14)$$

$$y_r \in \{0.1\} \qquad \forall r \in Rs \qquad (15)$$

This parameter depends on the number of routes with a reduced cost less than zero obtained from the slave problem and is 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 N (Eq. (14)). The objective function is to minimize the total distance of the routes (Eq. (12)). Once the master problem is solved, it generates $\pi + 1$ dual variables (W_i^t) used by the slave problem in the objective function.

3.2.2 Slave Problem (AP)

For the slave problem, a CSP is performed, where the objective is to minimize the reduced cost of the original problem (16). We want to achieve as many routes as possible until we obtain a route with a value greater than or equal to σ in the objective function value by using the binary variable x_{ij}, which is defined as a binary variable that takes the value of 1 if the arc linking customer $i \epsilon C$ with customer $j \epsilon C$ is activated. Additionally, the remaining constraints that were not utilized in the master problem are maintained for this problem. Constraints (17) and (18) ensure that each customer is visited strictly once. Constraints (19) and (20) ensure that the route starts and ends at the depot, while constraints (21) ensure that if an arc is activated ($x_{ij} = 1$) towards a customer, j must also originate from node ($x_{ij} = 1$). Additionally, the volume and weight of the truck cannot be exceeded. This limitation is achieved by constraints (22) and (23), respectively. An iterative system of cuts is implemented, where the customers generate sub-tours using customers (24). The number of active nodes of the route is restricted to maintain it without sub tours. In this way, the model is reoptimized until there are no subtours in the solution. The nature of the variables is described by (25).

$$\min \sum_{i \in C} \sum_{j \in C} (d_{ij} - \pi_i^t) x_{ij}\, \sigma \qquad (16)$$

s.t:

$$\sum_{j \in C} x_{ij} = 1 \qquad \forall i \in C \qquad (17)$$

$$\sum_{i \in C} x_{ij} = 1 \qquad \forall j \in C \qquad (18)$$

$$\sum_{j \in C} x_{0j} = 1 \qquad (19)$$

$$\sum_{i \in C} x_{i0} = 1 \qquad (20)$$

$$\sum_{i \in C} x_{ih} - \sum_{j \in C} x_{hj} = 0 \qquad h \in C \qquad (21)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ij}\, v_i \leq VolC \qquad (22)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ij}\, p_i \leq pesC \qquad (23)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijr} \leq |S| - 1 \qquad \forall\ |S| \geq 1. S \subseteq V \qquad (24)$$

$$x_{ij} \in \{0.1\} \qquad \forall\ i.j \in C \qquad (25)$$

Once the shortest route problem is solved, parameter a_{ir} (26) for the master problem is obtained.

$$a_{ir} = \sum_{j \in C} x_{ij} \qquad \forall\ i \in C;\ r \epsilon R \text{ generated} \qquad (26)$$

3.3. Packing validator

In this work, a GRASP algorithm based on the idea proposed by Martínez et al. (2015) was employed as a validation routine for the packing constraints. This algorithm is developed to solve the problem of three-dimensional packing considering the constraints of multiple destinations. In this way, when a solution is obtained for the slave problem, the GRASP algorithm is called to validate if the obtained column meets the packing constraints. If it is a valid solution, it becomes a new column of the Master Problem. Otherwise, this column is discarded, and the slave problem must be solved again until a valid column is obtained.

Algorithm 2. Constructive Algorithm

Parameters: *A*, *N*, *deterministic:* Boolean flag to select the deterministic or random version;
Input: **List** B_i: list of boxes of the customer ***i*****, List** *E*: current empty spaces, **Packing Pattern** *P*: current packing pattern, *k*: current iteration;
Output: Packing Pattern *P*, **List** *E*, **List** B_i.

```
while B_i ≠ ∅ and E ≠ ∅ do
    e ← SelectMaximalSpace(E)
    List Layers ← GenerateLayersList(e, B_i)
    if deterministic = false then
        List RCL ← BuildRCL(A, N; Layers, k)
        Layer L ← SelectLayerRandomly(RCL)
    else
        Layer L ← SelectLayerBestFitsOnSpace(Layers, e)
    end if
    P ← LocateLayerOnSpace(L, e)
    E ← UpdateListOfMaximalSpaces(P, e, E)
    B_i ← UpdateListOfRemainingBoxes(L)
end while
return P, E, B_i
```

The reactive GRASP algorithm consists of two phases to verify the route's viability. The first phase considers a constructive algorithm, and the second is an improvement phase. In the constructive phase, a solution is created by adding the items pseudorandomly. In the second phase, the algorithm performs improvement movements that consist of emptying and filling again the vehicle but in a deterministic fashion. Algorithm 2 shows the difference between phases, for the constructive phase, the flag deterministic has a value false, producing pseudo-random assignations of items to the empty spaces, the degree of randomness varies depending on the current iteration (*k*), the number of iterations for training the alphas values (*N*), the set of alpha values selected (*A*), and the performance obtained during the training process. A detailed explanation of the configuration of the coefficients for the reactive feature is presented in Cuellar-Usaquén et al. (2022).

Algorithm 3. GRASP Algorithm

Parameters: *TotalIter*: total GRASP iterations, *A*: set of α coefficients for the reactive feature, *N*: number of iterations for α training, *K*: percentage of removed boxes;
Input: **List** *R*: route**, List** *B*: list of boxes of each customer in the route *R***, Space** *v*: dimensions of the vehicle;
Output: **Boolean** *Packable*.

```
1: for k ← 1 to TotalIter do
2:     List E ← CreateMaximalSpace(v)
3:     Packing Pattern P ← ∅
2. 4:      for each customer i in R do
5:         P, E, B_i ← ConstructiveAlgorithm(A, N, false; B_i, E, P, k)
6:         if B_i ≠ ∅ then
7:             E ← UpdateListOfMaximalSpacesForNextCustomer(P, E)
8:         else
9:             P, E, B_i ← RemoveK%(B_i, E, P, K)
10:            P, E, B_i ← ConstructiveAlgorithm(A, N, true; B_i, E, P, k)
11:            if B_i ≠ ∅ then
12:                E ← UpdateListOfMaximalSpacesForNextCustomer(P, E)
13:            else
14:                break for
15:                next k and goto line 1:
16:            end if
17:        end if
18:    end for
19:    return Packable = true
20: end for
21: return Packable = false
```

Algorithm 3 details the main steps of the GRASP; it consists of an iterative process (lines 1-20); in each iteration, all the boxes are trying packed, as soon all boxes can be packed the iterating process is finished (line 19), but if the iterations are exhausted then the algorithm returns saying that the route is not packable (line 21). Thus, each customer is intended to pack inside the remaining space (lines 4-18), using the scheme of construct and improved (lines 5-12), when a mid-route customer cannot be packed, all the iteration is discarded (line 15). The highlighting issues of the GRASP are listed as follows:

I. A random strategy based on maximizing the use of the box spaces that best fit into the available (empty) space is applied. This process consists of five 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 customer is chosen, and its items are packed depending on the increased occupied volume that best fits into the container. The list of spaces is updated in the third step, iteratively generating new parallelepipeds for the current customer every time a new item is added. The list of parallelepipeds of the remaining spaces for the next customer is updated.

II. The randomization procedure allows generating combinations of packing with the selected items and occupying as much space as possible. This outcome is achieved by creating combinations of the boxes' packing and then randomly generating the scenarios based on the dimensions of the boxes and possible orientations of packing.

III. The improvement of the current solutions allows mobilizing and compressing the load to increase the occupied space. The first part of the algorithm uses a constructive deterministic method to repack the unpacked and removed boxes based on the criterion of "best item that fits." The second part of the improved algorithm performs the previously mentioned method to obtain a partial solution when the occupied space decreases.

The GRASP packing solution indicates whether the generated column (*R* route) could be packed. If the route could be packed, the results are sent to the master problem and the next iteration of the generation of columns. Otherwise, if the route is not feasible by packing, the AP is reoptimized by adding a cut that prohibits the obtained column (27). This process is repeated until the packaging algorithm accepts a new solution.

$$\sum_{i \in CR} \sum_{j \in CR} x_{ij} \leq |n|; \text{ where } CR \text{ and } n \text{ are the ids and number of customers in the route } R \quad (27)$$

3.4 Multi-Start Randomized Constructive Algorithm (MSRCA)

A MSRCA was employed to initialize the master problem, which consists of 4 steps:

I. The list of vehicles and customers is initialized.

II. A vehicle (V_s) is selected depending on one of the following two criteria, with a probability of 50% each:
   - Vehicle with less remaining volume (0% - 50%)
   - Vehicle with less remaining weight (51% - 100%)

   According to the chosen criterion, the vehicles are ordered, and a pseudorandom probability is assigned to each vehicle (see Equation 28). One vehicle is selected randomly.

III. Choosing the customer is performed in a similar way to choosing the vehicles. A customer is chosen (C_s) based on two criteria with a probability of 50% each:
   - Customer with the highest volume (0% - 50%)
   - Customer with better weight adjustment (51% - 100%)

   Customers that can be packed are ordered according to the chosen criterion. A pseudorandom probability is assigned to each customer (see Equation 28). If there are no customers on the list, the vehicle is closed, and we return to step 2.

IV. The list of customers and remaining vehicles is updated, and the V_s route is updated with the C_s. In addition, if the V_s' available volume is less than the minimum of the remaining customers' volumes or if the V_s' available weight is less than the remaining customers' minimum weight, the vehicle is closed, and it returns to step 2. If there are no vehicles available, but if the remaining customers are returned to step 1, in the opposite case where vehicles and customers remain, return to step 2.

$$Pseudo - Random\ Prob = \frac{\left[\frac{Size\ List - Position\ Candidate + 1}{Size\ List}\right]}{0{,}5(Size\ List + 1)} \quad (28)$$

An example of the pseudorandom probability is presented as follows: We consider three homogeneous vehicles in weight and volume (a, b, and c). For an intermediate iteration, truck a has 30 units of remaining volume, truck b has 50 units of remaining volume, and truck c has 20 remaining volume units. In step 2, with a probability of 0.45, the vehicle's criterion with the least volume is chosen. The trucks' final results obtain c, a, and b by using (28).

$$Pseudo - Random\ Prob(Truck\ a) = \frac{\left[\frac{3-2+1}{3}\right]}{0{,}5(3+1)} = \frac{2}{6} \quad (29)$$

$$Pseudo - Random\ Prob(Truck\ b) = \frac{\left\lceil\frac{3-3+1}{3}\right\rceil}{0,5(3+1)} = \frac{1}{6} \quad (30)$$

$$Pseudo - Random\ Prob(Truck\ c) = \frac{\left\lceil\frac{3-1+1}{3}\right\rceil}{0,5(3+1)} = \frac{1}{2} \quad (31)$$

Therefore, from 0% to 50% probability, the chosen truck is c, from 51% to 83.3% probability, the chosen truck is b, and from 83.4% to 100% probability, the chosen truck is a. From a random number of 0.38 results, vehicle c is chosen. In this way, the MSRCA algorithm was utilized to obtain the routes initializing the a_{ir}parameter of the master problem.

4. Computational results

The MSRCA algorithm was coded on a computer with the following specifications: Windows 7 Enterprise ®, with an Intel ® Core ™ i7-4610 M CPU @ 3.0 GHz and 16 GB of RAM. The column generation for the CVRP and GRASP algorithm was coded on a computer with the following specifications: Lenovo Legion Y520 – Linux Ubuntu 16.04 LTS - Intel® Core ™ i7-7700HQ CPU @ 2.80 GHz × 8 and 15.5 GB of RAM. The optimization algorithm is IBM CPLEX Studio 12.6®, and the C ++ programming language has been chosen. This study uses the value of the number of iterations proposed by Escobar et al. (2015). The proposed approach has considered the classical set of benchmark instances (27 instances) to validate the performance of the proposed approach. The former has been compared with the best-known published approaches for the 3L–CVRP. Several best-known results have been reached. The computing time of the proposed methodology is quite high compared to the published approaches. The benchmark set for the 3L–CVRP has been taken from the library published in http://or.dei.unibo.it/instances/three-dimensional-capacitated-vehicle-routing-problem-3l-cvrp. On the other hand, given that the slave problem is not necessarily solved to optimality, it was necessary to calibrate the maximum number of obtained solutions. An experimental value of a maximum of six complete solutions was obtained.

Table 1
Obtained results by the MSRCA Heuristic Algorithm

Instance	Customers	Items	Vehicles	Time (Seconds)
1	15	32	4	<1
2	15	26	5	<1
3	20	37	4	6
4	20	36	6	<1
5	21	45	6	<1
6	21	40	6	<1
7	22	46	6	1
8	22	43	6	<1
9	25	50	8	2
10	29	62	8	1
11	29	58	8	1
12	30	63	9	<1
13	32	61	8	1
14	32	72	9	1
15	32	68	9	1
16	35	63	11	<1
17	40	79	14	<1
18	50	99	12	4
19	71	147	18	9
20	75	155	17	12
21	75	146	18	9
22	75	150	17	7
23	75	143	16	19
24	100	193	22	25
25	100	199	26	15
26	100	198	23	13
Average				5.2

Source: Owner

The obtained results of the proposed approach are shown in Table 1 and Table 2. For Table 1, the first columns indicate the number of instances, number of customers, items, and vehicles. In the last column, the duration time is described in seconds, and the MSRCA is applied to obtain the solution of the routes for each instance, demonstrating its high effectiveness. Table 2 shows the summary for the best obtained results. We have defined independent parameters whose values must be determined by extensive computational experiments for the proposed approach. A calibration process has been carefully performed because the proposed approach's performance depends on the value for each parameter. This procedure is iteratively performed by considering every single factor (variable) and finding its "best value." The initial values of some parameters are obtained from previous similar works for some variants of routing and packing problems. In this way, a comparative analysis of the efficiency and quality of the solution was carried out. The other parameters were adjusted by implementing extensive computational tests, fixing the operator that generates the best quality solutions. Based on the test results, we examine the parameters sequentially according to their a priori importance and attempt to find the "best" treatment for each factor.

Table 2
Obtained results of proposed approach

Instance	Customers	Items	Vehicles	Proposed Approach	Solution Time(s)†
1	15	32	4	300.70	185
2	15	26	5	334.96	1
3	20	37	4	392.63	79
4	20	36	6	448.48	5
5	21	45	6	443.61	138
6	21	40	6	498.16	149
7	22	46	6	769.68	92
8	22	43	6	884.32	209
9	25	50	8	641.23	6
10	29	62	8	820.35	57
11	29	58	8	804.32	1242
12	30	63	9	624.24	3
13	32	61	8	2645.95	157
14	32	72	9	1482.88	233
15	32	68	9	1341.14	79
16	35	63	11	698.61	1
17	40	79	14	824.01	8
18	50	99	12	762.35	105
19	71	147	18	604.65	275
20	75	155	17	1168.20	1557
21	75	146	18	1150.32	7160
22	75	150	17	1130.54	925
23	75	143	16	1165.43	834
24	100	193	22	1464.87	670
25	100	199	26	1600.35	725
26	100	198	23	1622.20	2138

Source: Owner

Table 3 shows the obtained results on the benchmarking set comparing with the previous published papers. The first columns describe the number of nodes, items, and vehicles for each instance. A set of instances of the specialized literature has been employed to validate the efficiency of the AHBGC algorithm. The obtained solution and computational time necessary to reach it were taken into account. The obtained results are compared with the best solutions obtained in previous works. We have considered the best solution (BKS) published by the literature's different previous works. Four representative works are shown, each with its respective total distance of the routes, GAP concerning the BKS, and computing time (seconds) to obtain the solution. In the last columns, the obtained solution of the former algorithm is shown. In addition, Table 2 shows the average time and average GAP (%) of the published works concerning the BKS values and relative GAP (%) concerning the BKS. For the applied instances, the MSRCA algorithm identified a feasible solution for 3D-BPP, where the large problem had 26 boxes and 15 customers, while the most complex problem had 199 boxes and 100 customers. As shown in Table 2, the algorithm has very competitive results regarding the objective function and time to obtain solutions, with an average of 655 seconds for the 26 instances.

The proposed approach has been tested in 26 instances of the literature, where the small problem has 15 customers and 4 vehicles, and the large problem has 100 customers and 26 vehicles (Table 2). The results show stable behavior, although when increasing the number of customers, there is an overflow of distance and time. As the number of customers increases, the slave problem presents great inconveniences due to its complexity. It is important to note that competitive results have been obtained in fewer than 50 customers. The average computing time is still very competitive, despite analyzing the routes' total distances. On the other hand, the GAP percentage concerning the BKS obtains a value of 2.26 %, and although it is not the best solution if competitive results are presented, it will serve as a reference for future work.

Table 3
Comparison results of published algorithms for 3L–CVRP

Instance	BKS	Gendreau et al. (2006)	GAP BKS (%)	Solution Time (s)	Tarantilis et al. (2009)	GAP BKS (%)	Solution Time (s)	Bortfeldt (2012)	GAP BKS (%)	Average Time (s)	Ruan, et al. (2013)	GAP BKS (%)	Solution Time (s)	Escobar, et al. (2015)	GAP BKS (%)	Solution Time (s)	Proposed Approach	GAP BKS (%)	Solution Time(s)†
1	300.70	316.32	5.20	1800	321.47	6.91	13	302.02	0.44	72	303.21	0.84	99	**300.70**	0.00	107	**300.70**	0.00	185
2	334.96	350.58	4.66	1800	**334.96**	0.00	12	**334.96**	0.00	1	**334.96**	0.00	5	340.55	1.67	7	**334.96**	0.00	1
3	392.63	447.73	14.03	1800	430.95	9.76	541	**392.63**	0.00	182	398.05	1.38	94	404.03	2.90	154	**392.63**	0.00	79
4	430.89	448.48	4.08	1800	458.04	6.30	324	437.19	1.46	16	440.68	2.27	47	**430.89**	0.00	8	448.48	4.08	5
5	443.61	464.24	4.65	1800	465.79	5.00	100	**443.61**	0.00	183	452.56	2.02	64	492.24	10.96	456	**443.61**	0.00	138
6	498.16	504.46	1.26	1800	507.96	1.97	1212	**498.16**	0.00	24	498.56	0.08	197	498.32	0.03	17	**498.16**	0.00	149
7	769.68	831.66	8.05	1800	796.61	3.50	365	**769.68**	0.00	133	790.23	2.67	317	789.78	2.61	335	**769.68**	0.00	92
8	810.89	871.77	7.51	1800	880.93	8.64	230	**810.89**	0.00	139	820.67	1.21	99	875.08	7.92	496	884.32	9.06	209
9	630.13	666.1	5.71	1800	642.22	1.92	982	**630.13**	0.00	24	635.50	0.85	353	639.26	1.45	78	641.23	1.76	6
10	820.35	911.16	11.07	3600	884.74	7.85	1308	**820.35**	0.00	175	836.21	1.93	411	829.23	1.08	2063	**820.35**	0.00	57
11	762.51	819.36	7.46	3600	873.43	14.55	523	803.61	5.39	136	825.75	8.29	198	**762.51**	0.00	1370	804.32	5.48	1242
12	614.59	651.58	6.02	3600	624.24	1.57	295	**614.59**	0.00	14	626.59	1.95	89	641.30	4.35	62	624.24	1.57	3
13	2645.95	2928.34	10.67	3600	2799.74	5.81	2193	**2645.95**	0.00	268	2739.80	3.55	320	2759.12	4.28	2017	**2645.95**	0.00	157
14	1368.42	1559.64	13.97	3600	1504.44	9.94	4581	**1368.42**	0.00	312	1469.38	7.38	268	1482.88	8.36	3599	1482.88	8.36	233
15	1341.14	1452.34	8.29	3600	1415.42	5.54	2528	**1341.14**	0.00	312	1369.69	2.13	357	1374.22	2.47	3598	**1341.14**	0.00	79
16	698.61	707.85	1.32	3600	**698.61**	0.00	4257	**698.61**	0.00	3	703.15	0.65	432	703.38	0.68	11	**698.61**	0.00	1
17	824.01	920.87	11.75	3600	872.79	5.92	2096	866.4	5.14	3	872.05	5.83	375	871.63	5.78	23	**824.01**	0.00	8
18	741.74	871.29	17.47	7200	818.68	10.37	2509	**741.74**	0.00	417	780.37	5.21	326	838.45	13.04	3613	762.35	2.78	105
19	587.95	732.12	24.52	7200	641.57	9.12	1941	**587.95**	0.00	427	605.59	3.00	1375	634.27	7.88	3601	604.65	2.84	275
20	1090.22	1275.2	16.97	7200	1159.72	6.37	2823	**1090.22**	0.00	443	1119.45	2.68	1337	1147.64	5.27	3599	1168.20	7.15	1557
21	1147.80	1277.94	11.34	7200	1245.35	8.50	2686	**1147.80**	0.00	424	1167.28	1.70	1248	1218.54	6.16	3599	1150.32	0.22	7160
22	1130.54	1258.16	11.29	7200	1231.92	8.97	4659	**1130.54**	0.00	426	1171.77	3.65	1295	1133.71	0.28	3598	**1130.54**	0.00	925
23	1116.13	1307.09	17.11	7200	1201.96	7.69	4854	**1116.13**	0.00	411	1136.27	1.80	1106	1189.25	6.55	3598	1165.43	4.42	834
24	1407.36	1570.72	11.61	7200	1457.46	3.56	5726	**1407.36**	0.00	453	1426.34	1.35	2001	1464.03	4.03	3598	1464.87	4.09	670
25	1585.46	1847.95	16.56	7200	1711.93	7.98	6283	1600.35	0.94	431	**1585.46**	0.00	1459	1664.84	5.01	3599	1600.35	0.94	725
26	1529.86	1747.52	14.23	7200	1646.44	7.62	9916	**1529.86**	0.00	435	1562.18	2.11	3140	1647.88	7.71	3600	1622.20	6.04	2138
Average Time (s)				4223			2421			226			654			1800			655
Average GAP (%)			10.26			6.36			0.51			2.48			4.25			2.26	

Source: Owner

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

† All computing times are reported in seconds.

5. Conclusing remarks and future work

This paper has presented a novel heuristic solution based on the generation of columns for vehicle routing and packaging of goods, which finds feasible routes in terms of volume, weight, and load inside the truck and minimizes the distances of these routes. In addition, the formulation and different algorithms used to perform the work have been presented. Additionally, the results were presented for different instances of the literature. Regarding the model's behavior, it can be inferred that the model presents competitive results on a small scale. In contrast, there is an overflow of time and distance on a large scale because the slave problem is an NP-hard problem. Finding optimal solutions for this model becomes very difficult. Therefore, the AHBGC algorithm's performance depends on the AP model's structure, given by the number of customers to address the problem. In future work, we propose acceleration methods for the slave problem. In addition, an algorithm to complement the column generation approach must be developed. Indeed, since a new column is added to the master problem, it is impossible to ensure that it belongs to the base by considering a favorable reduced cost. The new approach must help the solution enter the base. On the other hand, a new column control procedure must be considered to remove it from the routing matrix and decrease its size. Additionally, stochastic considerations with appropriate solution techniques must be considered, such as those proposed by Escobar (2012), Escobar et al. (2012), Escobar et al. (2013a), and Paz et al. (2015), for related problems. In addition, metaheuristic algorithms based on granular search space could be extended to related routing problems (Escobar et al., 2014a; Escobar et al., 2013b; Bernal et al., 2021; Linfati et al., 2014; Escobar et al., 2014b).

References

Bernal, J., Escobar, J. W., Paz, J. C., Linfati, R., & Gatica, G. (2018). A probabilistic granular tabu search for the distance constrained capacitated vehicle routing problem. *International Journal of Industrial and Systems Engineering, 29*(4), 453-477.

Bernal, J., Escobar, J. W., & Linfati, R. (2021). A simulated annealing-based approach for a real case study of vehicle routing problem with a heterogeneous fleet and time windows. *International Journal of Shipping and Transport Logistics, 13*(1-2), 185-204.

Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research, 39*(9), 2248-2257.

Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second—a heuristic for the vehicle routing and loading problem. *Computers & Operations Research, 40*(3), 873-885.

Bortfeldt, A., Hahn, T., Männel, D., & Mönch, L. (2015). Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints. *European Journal of Operational Research, 243*(1), 82-96.

Cuellar-Usaquén, D., Gomez, C. and Álvarez-Martínez, D. (2022), A GRASP/Path-Relinking algorithm for the traveling purchaser problem. *International Transaction in Operations Research*. https://doi.org/10.1111/itor.12985

Escobar, J. W. (2012). Rediseño de una red de distribución con variabilidad de demanda usando la metodología de escenarios. *Revista Facultad de Ingeniería, 21*(32), 9-19.

Escobar, J. W., Bravo, J. J., & Vidal, C. J. (2012). Optimización de redes de distribución de productos de consumo masivo en condiciones de riesgo. In Proceedings of XXXIII Congreso Nacional de Estadística e Investigación Operativa (SEIO), Madrid, Spain.

Escobar, J. W., Bravo, J. J., & Vidal, C. J. (2013a). Optimización de una red de distribución con parámetros estocásticos usando la metodología de aproximación por promedios muestrales. *Revista Científica Ingeniería y Desarrollo, 31*(1), 135-160.

Escobar, J. W., Linfati, R., & Toth, P. (2013b). A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research, 40*(1), 70-79.

Escobar, J. W., Linfati, R., Toth, P., & Baldoquin, M. G. (2014a). A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of heuristics, 20*(5), 483-509.

Escobar, J. W., Linfati, R., Baldoquin, M. G., & Toth, P. (2014b). A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem. *Transportation Research Part B: Methodological, 67*, 344-356.

Escobar, L. M., Martínez, D. Á., Escobar, J. W., Linfati, R., & Mauricio, G. E. (2015). A hybrid metaheuristic approach for the capacitated vehicle routing problem with container loading constraints. In 2015 International Conference on Industrial Engineering and Systems Management (IESM) (pp. 1374-1382). IEEE.

Escobar-Falcón, L. M., Álvarez-Martínez, D., Granada-Echeverri, M., Escobar, J. W., & Romero-Lázaro, R. A. (2016). A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3L-CVRP). *Revista Facultad de Ingeniería Universidad de Antioquia, 78*, 09-20.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research, 201*(3), 751-759.

Garey, M. R. & Johnson, D. S (1979). *Computers and Intractability: A guide to the theory of NP completeness*, W. H. Freeman and co., New York.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science, 40*(3), 342-350.

Hokama, P., Miyazawa, F. K., & Xavier, E. C. (2016). A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications, 47*, 1-13.

Junqueira, L., Oliveira, J. F., Carravilla, M. A., & Morabito, R. (2013). An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research, 20*(5), 645-666.

Junqueira, L., & Morabito, R. (2015). Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier. *Computers & Industrial Engineering, 88*, 110-130.

Lacomme, P., Toussaint, H., & Duhamel, C. (2013). A GRASP× ELS for the vehicle routing problem with basic three-dimensional loading constraints. *Engineering Applications of Artificial Intelligence, 26*(8), 1795-1810.

Linfati, R., Escobar, J. W., & Cuevas, B. (2014). An algorithm based on granular tabu search for the problem of balancing public bikes by using multiple vehicles. *Dyna, 81*(186), 284-294.

Linfati, R., & Escobar, J. W. (2018). Reoptimization heuristic for the capacitated vehicle routing problem. *Journal of Advanced Transportation*, 2018.

Mahvash, B., Awasthi, A., & Chauhan, S. (2017). A column generation based heuristic for the capacitated vehicle routing problem with three-dimensional loading constraints. *International Journal of Production Research, 55*(6), 1730-1747.

Mak-Hau, V., Moser, I., & Aleti, A. (2018). *An exact algorithm for the heterogeneous fleet vehicle routing problem with time windows and three-dimensional loading constraints*. In Data and Decision Sciences in Action (pp. 91-101). Springer, Cham.

Männel, D., & Bortfeldt, A. (2016). A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. *European Journal of Operational Research, 254*(3), 840-858.

Martínez, D. A., Alvarez-Valdes, R., & Parreño, F. (2015). A grasp algorithm for the container loading problem with multi-drop constraints. *Pesquisa Operacional, 35*(1), 1-24.

Miao, L., Ruan, Q., Woghiren, K., & Ruo, Q. (2012). A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. *RAIRO-Operations Research-Recherche Opérationnelle, 46*(1), 63-82.

Pace, S., Turky, A., Moser, I., & Aleti, A. (2015). Distributing fibre boards: a practical application of the heterogeneous fleet vehicle routing problem with time windows and three-dimensional loading constraints. *Procedia Computer Science, 51,* 2257-2266.

Paz, J., Orozco, J., Salinas, J., Buriticá, N., & Escobar, J. (2015). Redesign of a supply network by considering stochastic demand*. International Journal of Industrial Engineering Computations, 6*(4), 521-528.

Pinto, T., Alves, C., & de Carvalho, J. V. (2015). Exploring a Column Generation Approach for a Routing Problem with Sequential Packing Constraints. In Operations Research and Big Data (pp. 159-166). Springer, Cham.

Ruan, Q., Zhang, Z., Miao, L., & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research, 40*(6), 1579-1589.

Tao, Y., & Wang, F. (2010). A new packing heuristic based algorithm for vehicle routing problem with three-dimensional loading constraints. *In 2010 IEEE International Conference on Automation Science and Engineering (pp. 972-977)*. IEEE.

Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems, 10*(2), 255-271.

Vega-Mejía, C., González-Neira, E., Montoya-Torres, J., & Islam, S. (2020). Using a hybrid heuristic to solve the balanced vehicle routing problem with loading constraints. *International Journal of Industrial Engineering Computations, 11*(2), 255-280.

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research, 265*(3), 843-859.

Zhang, Z., Wei, L., & Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological, 82*, 20-35.