WORKING PAPER SERIES

Solving the Pickup and Delivery Problem with 3D Loading Constraints and Reloading Ban

Dirk Männel/Andreas Bortfeldt

Working Paper No. 16/2015

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

FACULTY OF ECONOMICS
AND MANAGEMENT

Impressum (§ 5 TMG)
Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
Dirk Männel and Andreas Bortfeldt
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Solving the Pickup and Delivery Problem with 3D Loading Constraints and Reloading Ban

Dirk Männel[1], Andreas Bortfeldt[1*]
[1] Otto von Guericke University, Universitätsplatz 2, 39106 Magdeburg, Germany
andreas.bortfeldt@ovgu.de, dirk.maennel@gmx.de
[*] Corresponding author, phone: 0049 391 6711842

Abstract

In this paper, we extend the classical Pickup and Delivery Problem (PDP) to an integrated routing and three-dimensional loading problem, called PDP with 3D loading constraints (3L-PDP). A set of routes of minimum total length has to be determined such that each request is transported from a loading site to the corresponding unloading site. In the 3L-PDP, each request is given as a set of 3D rectangular items (boxes) and the vehicle capacity is replaced by a 3D loading space. This paper is the second one in a series of articles on 3L-PDP. In both articles we investigate which constraints will ensure that no *re*loading effort will occur, i.e. that no box is moved after loading and before unloading. In this paper, the focus is laid on the so-called reloading ban, a packing constraint that ensures identical placements of same boxes in different packing plans. We propose a hybrid algorithm for solving the 3L-PDP with reloading ban consisting of a routing and a packing procedure. The routing procedure modifies a well-known large neighborhood search for the 1D-PDP. A tree search heuristic is responsible for packing boxes. Computational experiments were carried out using 54 3L-PDP benchmark instances.

Key words: Transportation, vehicle routing, pickup and delivery, 3D loading constraints.

1 Introduction

The classical Pickup and Delivery Problem (PDP) is an important type of Vehicle Routing Problems (VRPs) with many applications in mixed cargo transportation (Parragh et al., 2008). A set of transportation requests is given and each of them is characterized by a (1D) demand, a specific loading site (pickup point) and a specific unloading site (delivery point). All requests have to be served by a fleet of homogeneous vehicles with uniform (1D) capacity. A set of routes, each starting and ending at a single depot, has to be found so that each request is served at one route that visits its loading site before its unloading site. The capacity of used vehicles must never be exceeded by the loaded goods, and the transportation cost, given by the total travel distance, should be minimized. Moreover, the length of each route as well as the number of routes must not exceed a given limit.

In recent years, more and more VRPs were formulated and solved as combined routing and 3D loading problems, i.e. 1D customer demands were replaced by sets of parallelepipeds (boxes) and 3D rectangular loading spaces are substituted for 1D capacities of vehicles. This was first done by Gendreau et al. (2006) for the capacitated vehicle routing problem (CVRP), resulting in the CVRP with 3D loading constraints (3L-CVRP). Each constructed route of a VRP solution has now to be accompanied by a packing plan for the boxes that are loaded and unloaded on that route. This essential modification allows for a more detailed and realistic modeling of mixed cargo transportation by vehicles (cf. Bortfeldt and Homberger, 2013).

In the paper at hand, we extend the classical PDP to an integrated routing and 3D loading problem, called hereafter PDP with three-dimensional loading constraints (3L-PDP). This paper is the second one in a series of papers on the 3L-PDP (see Männel and Bortfeldt, 2015). To make the paper self-contained several considerations of the first paper are repeated.

Our main concern in the problem formulation of 3L-PDP is to guarantee that in 3L-PDP solutions any *re*loading effort is excluded. That is, the boxes should not be moved *after* they were loaded and *before* they are unloaded. In the 3L-CVRP, this is ensured by the so-called last-in-first-out (LIFO) constraint. However, in the 3L-PDP, we must introduce further constraints to eliminate any reloading effort. It turns out that we can choose between a routing constraint (called independent partial routes (IPR) constraint) and a packing constraint (called reloading ban) to achieve that purpose.

In our former paper, a hybrid algorithm was proposed for solving mainly the 3L-PDP variant with IPR constraint. This time, we develop a hybrid algorithm for solving the 3L-PDP with reloading ban. Again, the hybrid algorithm consists of the modified large neighborhood search (LNS) algorithm by Ropke and Pisinger (2006) for the 1D-PDP and the tree search (TRS) algorithm for packing boxes by Bortfeldt (2012). The hybrid algorithm is subjected to a numerical test carried out by means of 54 3L-PDP benchmark instances with up to 100 requests.

Regarding the relevant literature, we refer the reader to the review given in Männel and Bortfeldt (2015). Some of the most important metaheuristic solution methods for the 1D-PDP (which is NP-hard) were suggested by Li and Lim (2001), Bent and van Hentenryck (2006) and Ropke and Pisinger (2006). A survey paper on VRPs with loading constraints vehicle routing (which are NP-hard and difficult to solve) was written very recently by Pollaris et al. (2015). Metaheuristic solution methods for the 3L-CVRP were developed, e.g. by Gendreau et al. (2006), Tarantilis et al. (2009), Fuellerer et al. (2009), Bortfeldt (2012) and Tao and Wang (2015). Moura and Oliveira (2009) specified and solved the VRP with time windows and 3D loading constraints which was also addressed by Bortfeldt and Homberger (2013). Bortfeldt et al. (2015) proposed hybrid algorithms for solving the VRP with backhauls and 3D loading constraints. Bartók and Imreh (2011) specified a local search heuristic for solving a 3L-PDP variant without LIFO constraint, while Malapert et al. (2008) developed a heuristic for the PDP with 2D loading constraints (without reporting numerical results). One can state that the 3L-PDP and the 2L-PDP have not yet found sufficient attention.

The rest of the paper is organized as follows: Section 2 introduces several variants of the 3L-PDP that are described more formally in Section 3. Section 4 proposes the hybrid algorithm for solving the 3L-PDP with reloading ban. In Section 5 numerical results of experiments are presented and conclusions are drawn in Section 6.

2 Variants of 3L-PDP

It is assumed that vehicles are rear-loaded, i.e. boxes are loaded and unloaded at the rear and only by movements in length direction of the vehicle (see Figure 1). At the same time any reloading effort should be avoided, i.e. any repositioning and rotating of boxes after loading and before unloading. In the following, we specify sufficient conditions to rule out any reloading effort.

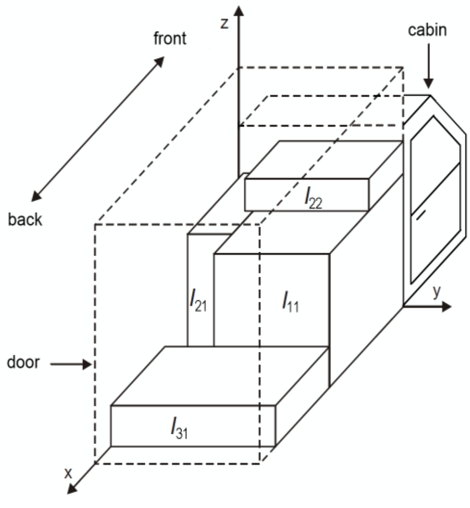

Figure 1: A loading space with placed boxes.

At first we must assume the request sequence (RS) constraint at delivery and pickup points of a route as well. The RS constraint can be thought of as extended LIFO constraint. At a delivery point, the RS constraint says that between a box A to be unloaded and the rear there is no box B to be unloaded later. Moreover, a box B to be unloaded later must not lie above box A. At a pickup point, the RS constraint requires that between a box A just loaded and the rear or above box A there is no box B that was loaded at an earlier pickup point. If the RS constraint would not be satisfied at a delivery or pickup point, boxes could not be unloaded or loaded by a pure movement in length direction and without moving other boxes. For a delivery point, placements of other boxes would have to be changed temporarily in order to unload boxes with this destination by pure length shifts. For a pickup point, placements of other boxes must be changed temporarily to reach the final positions for the loaded boxes by pure length movements. Thus, the RS constraint at delivery and pickup points is a necessary condition to avoid any reloading effort.

However, the RS constraint is not sufficient in this regard as the following consideration reveals. In a route for 3L-PDP, generally boxes of a request A are transported for a part of the route together with boxes of a request B and for another part together with boxes of a request C (and no longer with boxes of B) etc. Packing plans have to be provided for all parts of the route in which different sets of boxes are transported. If different packing plans are provided for the boxes of a request A, because the boxes are first to be packed with the boxes of request B and then with the boxes of request C the placements of the boxes of A may change. This would not necessarily violate required packing constraints. Thus, there would exist feasible 3L-PDP solutions including boxes that are to be *re*loaded after loading and before unloading; for an elaborated example see Männel and Bortfeldt (2015).

In order to rule out any reloading effort, we have to specify an extra constraint. There are two options to do so, i.e. we can introduce an additional packing constraint and, alternatively, we can define a routing constraint that rules out any reloading effort.

The additional packing constraint, termed reloading ban, requires that the placement of any box, including the position of a reference corner (or of the geometrical midpoint) and the spatial orientation of the box, must not undergo a (permanent) change after the box has been loaded and before the box is unloaded. The reloading ban is tailored to the general shape of 3L-PDP routes: it forbids explicitly a change of placements of boxes of a request A if they are loaded together with boxes of a request C after they have been loaded together with boxes of a request B.

The mentioned routing constraint, called independent partial routes (IPR) constraint, rules out any reloading effort by restricting the shape of the routes, i.e. in an implicit fashion. This is done by so-called 3L-PDP routing patterns, ensuring that the boxes of any request are not stored together with boxes of different requests in different parts of a route (see Männel and Bortfeldt, 2015).

Based on the above considerations, Table 1 shows a spectrum of five 3L-PDP variants. The RS constraint at loading sites is always required. The variants are specified by means of the RS constraint for unloading sites, the reloading ban and the IPR constraint. For each variant and constraint the entry is "y" if the constraint is to be met and "n" if not. If the IPR condition and the RS constraint at loading sites is required, RS constraint at unloading sites and reloading ban are automatically satisfied; this is marked by entry "a". In the last two columns, the expected reloading effort and the expected (total) travel distance are indicated. For example, if none of the three defining constraints must be observed, a high reloading effort is to be expected and the total travel distance will be very low. If in the opposite case the reloading effort is forced to be zero, the expected travel distance will be relatively high. If the reloading effort is ruled out by the IPR constraint, the total travel distance will be especially high as this constraint restricts the solution space more than the reloading ban.

Table 1: Five 3L-PDP variants (y: yes, n: no, a: automatically).

#	RS loading	RS unloading	Reloading ban	Independent partial routes	Reloading effort	Travel distance
1	y	n	n	n	high	very low
2	y	y	n	n	medium	low
3	y	n	y	n	medium	low
4	y	y	y	n	zero	medium
5	y	a	a	y	zero	high

In this paper, we will focus on the 3L-PDP variants including the reloading ban (variants 3 and 4), while the other variants were dealt with in Männel and Bortfeldt (2015).

3 Problem definition

Now we specify the 3L-PDP in a more formal fashion. We are given n requests each consisting of a loading site i, an unloading site $n+i$ and a set I_i of goods that are to be transported from i to $n+i$ (i = 1,…,n). There are v_{max} identical vehicles, originally located at the single depot (denoted by 0), with a rectangular loading space with length L, width W and height H. Let $V = \{0,1,\ldots,n,n+1,\ldots,2n\}$ be the set of all nodes, i.e. loading and unloading sites including the depot. Let E be a set of undirected edges (i,j) that connect all node pairs ($0 \leq i, j \leq 2n$, $i \neq j$) and let $G = (V, E)$ be the resulting graph. Let a travel cost c_{ij} ($c_{ij} \geq 0$) be assigned to each edge (i,j) and let the travel costs be symmetric, i.e. $c_{ij} = c_{ji}$ ($0 \leq i, j \leq 2n$, $i \neq j$). The set I_i includes m_i rectangular pieces (boxes) I_{ik} and the box I_{ik} has the length l_{ik}, the width w_{ik} and the height h_{ik} (i = 1,…,n, k = 1,…,m_i).

The loading space of each vehicle is embedded in the first octant of a Cartesian coordinate system in such a way that the length, width and height of the loading space lie parallel to the x, y, and z axes. The placement of a box I_{ik} in a loading space is given by the coordinates x_{ik}, y_{ik}, and z_{ik} of the corner of the box closest to the origin of the coordinates system; in addition, an orientation index o_{ik} indicates which of the possible spatial orientations is selected (i = 1,...,n, k = 1,...,m_i). A spatial orientation of a box is given by a one-to-one mapping of the three box dimensions and the three coordinate directions.

A packing plan P for a loading space comprises one or more placements and is regarded as feasible if the following three conditions hold: (FP1) each placed box lies completely within the loading space; (FP2) any two boxes that are placed in the same truck loading space do not overlap; (FP3) each placed box lies parallel to the surface areas of the loading space. Each vehicle is loaded and unloaded at the rear and empty at the beginning of a route.

A feasible route R is a sequence of $2p+2$ nodes ($p \geq 1$) that starts and ends at the depot. R should include the loading and unloading sites of p different (among the n given) requests and each loading site must precede the unloading site of the same request. A solution of the 3L-PDP is a set of v sequences (R_l, $P_{l,1},\ldots,P_{l,2p_l}$), where R_l is a route and $P_{l,q}$ is a packing plan ($l = 1,\ldots,v$, $q = 1,\ldots,2p_l$, p_l denotes the number of requests of route l).

$P_{l,q}$ represents the packing pattern of route l after having visited its $(q+1)th$ node, i.e. after some boxes were loaded or unloaded at the $(q+1)th$ node of route l. To be feasible, a solution must fulfil the following three conditions: (F1) all routes R_l and packing plans $P_{l,q}$ are feasible ($l = 1,\ldots,v$, $q = 1,\ldots,2p_l$); (F2) the loading site and the unloading site of each request occurs once in one route R_l ($l = 1,\ldots,v$); (F3) the packing plan $P_{l,q}$ for a route R_l and its $(q+1)th$ node contains placements exactly for those boxes to be loaded but not (yet) to be unloaded at the first $q+1$ nodes of the route.

In addition, the following routing and packing constraints are optionally to be satisfied:

(C1) *Request sequence constraint (RS constraint)*: A packed box b of a certain request is said to be in unloading position if there is no packed box b' of another request between b and the rear of the vehicle or above box b (cf. Figure 3). *Loading requirement* (C1-l): If the $(q+1)th$ node of route l is a loading site, then all boxes to be loaded there must be in unloading position in the packing plan $P_{l,q}$, i.e. after loading ($l = 1,\ldots,v$, $q = 1,\ldots,2p_l$). *Unloading requirement* (C1-u): If the $(q+1)th$ node is an unloading site, then all boxes to be unloaded there must be in unloading position in the packing plan $P_{l,q-1}$, i.e. before unloading ($l = 1,\ldots,v$, $q = 1,\ldots,2p_l$). This constraint ensures that all boxes of a given request can be loaded or unloaded exclusively by movements parallel to the longitudinal axis of the loading space of a vehicle and without moving boxes of other requests.

(C2) *Reloading ban*: Each box I_{ik} of request i must not be moved *after* loading and *before* unloading ($i = 1,\ldots,n$, $k = 1,\ldots,m_i$). If the box I_{ik} is loaded at the $(q+1)th$ node and unloaded at the $(q'+1)th$ node of route l, its placement (x_{ik}, y_{ik}, z_{ik}, o_{ik}) must be the same in the packing plans $P_{l,q}$, $P_{l,q+1},\ldots,P_{l,q'-1}$ ($i = 1,\ldots,n$, $k = 1,\ldots,m_i$, $l = 1,\ldots,v$, $1 \leq q < q' \leq 2p_l$).

(C3) *Independent partial routes constraint (IPR constraint)*: Each route R_l follows a routing pattern, i.e. it consists of one or more sub-patterns ($l = 1,\ldots,v$). A sub-pattern consists of a series of one or more loading sites (pickup points) followed by the corresponding unloading sites (delivery points) in inverse order.

(C4) *Weight constraint*: Each box I_{ik} has a positive weight d_{ik} ($i = 1,\ldots,n$, $k = 1,\ldots,m_i$) and the total weight of all boxes in a packing plan $P_{l,q}$ must not exceed a maximum load weight D ($l = 1,\ldots,v$, $q = 1,\ldots,2p_l$).

(C5) *Orientation constraint*: The height dimension of all boxes is fixed, while horizontal 90° turns of boxes are allowed. Thus, only two of six values are allowed for the orientation index o_{ik} of a placement ($i = 1,\ldots,n$, $k = 1,\ldots,m_i$).

(C6) *Support constraint*: If a box is not placed on the floor, a certain percentage a of its base area has to be supported by other boxes.

(C7) *Stacking constraint*: A fragility attribute f_{ik} ($i = 1,\ldots,n$, $k = 1,\ldots,m_i$) is assigned to each box. If a box is fragile ($f_{ik} = 1$), only other fragile boxes may be placed on its top surface, whereas both fragile and non-fragile boxes may be stacked on a non-fragile box ($f_{ik} = 0$).

(C8) *Route length constraint*: The total distance of a route must not exceed a specified maximum d_{max}. This constraint can also be understood as a route duration constraint if the vehicle velocity is set to a constant.

(C9) *Route number constraint*: The number of routes v must not exceed the number of vehicles v_{max}.

Finally, the 3L-PDP consists of determining a feasible solution that meets some of the constraints (C1) to (C9) and minimizes the total travel distance of all routes. More precisely, we consider the variants of 3L-PDP as specified in Table 1 (see above) and require constraints (C1) to (C3) in accordance to Table 1. The other constraints (C4) to (C9) are stipulated for each of the five variants of the 3L-PDP.

4 A hybrid algorithm for the 3L-PDP

In the following, we propose a hybrid algorithm for the 3L-PDP that is composed of a procedure for routing and one for packing. The routing procedure is derived from the adaptive LNS heuristic for solving the PDPTW by Ropke and Pisinger (2006). The TRS algorithm by Bortfeldt (2012) was further developed to specify a packing procedure that is able to observe the reloading ban. The following description is focused on the packing procedure, while other parts and aspects of the hybrid algorithm are described with more details in Männel and Bortfeldt (2015).

4.1 Routing procedure

The routing procedure is the superior module of the hybrid algorithm and is outlined in Figure 2. After an initial solution was specified, iterations of a neighborhood search are performed until a time limit is exceeded. Within each iteration, a number ξ of requests to be removed, a removal heuristic *Rh* and an insertion heuristic *Ih* are selected randomly. Several removal and insertion heuristics are available. The next solution is generated according to $s_{next} := Ih(Rh(s_{curr}, \xi))$, i.e. a set of ξ requests is removed from solution s_{curr} and then reinserted. Afterwards, it is tested whether s_{next} is accepted as new current solution s_{curr}. In this case, the best solution s_{best} is updated if necessary. Otherwise, the initial solution of the next iteration s_{curr} is not changed.

```
3l_pdp_lns (in: problem data, parameters, out: best solution s_best)
    construct initial solution s_curr and set s_best := s_curr
    while stopping criterion is not met do
        select number of requests to be replaced ξ, removal heuristic Rh and insertion heuristic Ih
        determine next solution: s_next := Ih(Rh(s_curr, ξ))
        check acceptance of s_next
        if s_next is accepted then
            s_curr := s_next
            if (f(s_curr) < f(s_best)) then s_best := s_curr endif
        endif
    endwhile
end.
```

Figure 2: LNS-based routing algorithm for the 3L-PDP.

The acceptance of solutions is tested by means of the well-known simulated annealing rule; thus, the search is embedded in an annealing process with a geometric cooling schedule. The selection probabilities for the removal and insertion heuristics are fix; i.e. a pure LNS is performed. Because of the limited number of vehicles some solutions may not include all requests. To cope with incomplete solutions, the concept of a virtual request bank is used (see Ropke and Pisinger, 2006, p. 2).

The removal and insertion heuristics are basically adopted from the original adaptive LNS heuristic and briefly summarized in Table 2. Besides the route number constraint (C9) also the weight constraint (C4) and the route length constraint (C8) are checked within the routing procedure, i.e. within the insertion heuristics. The initial solution is specified by the Regret-2 insertion heuristic starting with an empty solution. Further explanations regarding removal and insertion heuristics can be found in Ropke and Pisinger (2006) and in Männel and Bortfeldt (2015).

Table 2: Removal and insertion heuristics of the LNS heuristic for 3L-PDP.

Heuristic	Description
Random removal Rh_R	Removes iteratively requests that are selected at random.
Shaw removal Rh_S	Removes iteratively requests that are related in terms of location and weight.
Worst removal Rh_W	Removes iteratively a request whose removal leads to the largest cost (total travel distance) reduction.
Tour removal Rh_T	Removes all requests from a randomly chosen route. If less than ξ requests are removed in this way, further requests will be removed with Shaw removal.
Greedy insertion Ih_G	Inserts iteratively requests into the solution such that the increase of the cost function is minimal.
Regret-2 insertion Ih_{R2}	Inserts iteratively requests into the solution such that the gap in the cost function between inserting the request into its best and its second best route is maximal.
Regret-3 insertion Ih_{R3}	Inserts iteratively requests into the solution such that the sum of two gaps in the cost function is maximal. The first gap results from inserting the request into its best and its second best route, while the second gap results from inserting the request into its best and its third best route.

4.2 Integration of routing and packing

3D packing checks are incorporated in two parts of the routing procedure. On the one hand, they are integrated in all insertion heuristics (and called insertion packing checks then). If a new solution s_{next} is generated from an old one s_{curr}, a route of s_{curr} is modified mostly as some requests are removed and some requests are reinserted. Hence, it will suffice to perform packing checks in insertion heuristics that are carried out after removal heuristics.

However, it might also occur that only old requests are removed from a route. Sometimes the remaining boxes cannot be stored according to the old placements or in a feasible way at all. This is due to support (C6) and stacking (C7) constraint. Therefore, further packing checks are performed within the acceptance test of a solution s_{next} (see Figure 2) and called acceptance packing checks then. All routes of s_{next} are checked, among them those that resulted by a pure removing of requests. If there is no feasible packing plan for one site, the solution s_{next} will be discarded and the search continues with the last accepted solution. Hence, an accepted solution will have only feasible routes in terms of packing.

The insertion heuristics are applied to an incomplete solution s and a set of missing requests Rm. They implement iteratively a best insertion (as defined in Table 2) of an request $rq \in Rm$ into a route until s is complete or no further request can be inserted. In each iteration, a set of best insertions I_{best} is determined per missing request $rq \in Rm$ (the number of required insertions depends on the insertion heuristic and is, e.g. set to 1 for the greedy insertion). This is done by procedure select_best_insertions that is used by all insertion heuristics (see Figure 3).

The procedure select_best_insertions is organized in two parts. In the first part (*for*-loop), all potential insertions of a given request rq into any route of a solution s are provided. Each insertion must be feasible only in terms of route length (C8), route number (C9) and weight (C4). In 3L-PDP variants 3 and 4 no further constraints are checked here. Minimum cost insertions are collected in a list I_{cand}.

In the second part (*while*-loop), the insertions of I_{cand} are examined by ascending costs. In each cycle, the currently minimum cost insertion ins_{best} undergoes a 3D packing check, i.e. the insertion ins_{best} is applied to its route and the route is then checked in terms of the constraints (C1-l), (C1-u) and (C5) to (C7). If the outcome is positive, insertion ins_{best} is included into the set of best insertions I_{best}. Otherwise, the next cheapest insertion for the route of ins_{best} (if any) will replace ins_{best} in list I_{cand}. The procedure returns if I_{best} has n_{ins} insertions or if I_{cand} is empty.

The packing effort is kept low as the one-dimensional checks are carried out before 3D packing checks. Moreover, all possible insertions are first evaluated and sorted by cost *before* the "expensive" packing checks are made. By this technique, called "evaluating first, packing second", packing checks can be aborted each time after few 3D-feasible insertions have been detected.

```
select_best_insertions (in: solution s, request rq, no. of required insertions n_ins,
                        out: set of best rq-insertions I_best)
    I_best := Ø; list of insertion candidates I_cand := Ø
    for all routes r of solution s do
        I_route(r) := set of all 1D-feasible insertions of rq in route r
        sort I_route(r) by ascending cost
        if |I_route(r)| > 0 then I_cand := I_cand ∪ {I_route(r)(1)} endif    // add first insertion of I_route(r)
    endfor
    while |I_best| < n_ins and |I_cand| > 0 do
        sort I_cand by ascending cost
        best insertion ins_best := I_cand(1); I_cand := I_cand \ {ins_best}
        perform 3D packing check of insertion ins_best, i.e. of the resulting route
        if 3D packing check of ins_best successful then
            I_best := I_best ∪ {ins_best}    // next best insertion found
        else r := route of ins_best
            I_route(r) := I_route(r) \ {ins_best} // remove ins_best
            if |I_route(r)| > 0 then I_cand := I_cand ∪ {I_route(r)(1)} endif // add (new) first insertion
        endif
    endwhile
end.
```

Figure 3: Procedure select_best_insertions with packing check.

4.3 Packing checks

A 3L-PDP solution has to provide feasible packing plans for each route and each visited site per route. The plan for a site must include placements of all boxes already loaded and not yet unloaded after visiting this site. In order to reduce the effort spent for packing checks, we apply a similar methodology as in Männel and Bortfeldt (2015) to the 3L-PDP variants 3 and 4 (see Table 1):

- Additional constraints are formulated that are stronger than the RS constraints (C1-l) and (C1-u).
- It is shown that feasible packing plans for all sites of a route can be derived from feasible packing plans for selected pickup points of this route if the latter plans meet the additional (as well as original) constraints.
- While the additional constraints lead to a further restriction of the search space, the search becomes less costly as independent packing plans are to be provided only for few sites of a route.

We define a sequence of open pickup points (SOPP) as a sequence of pickup points within a route of a 3L-PDP solution with following characteristics: (i) the last point of the sequence is followed by a delivery point in the route; (ii) the sequence contains all and only pickup points of the route whose delivery points lie behind the last sequence point.

Let m_2 ($m_2 \geq 1$) be the number of consecutive pickup points lying at the end of the sequence. Let m_1 ($m_1 \geq 0$) be the number of pickup points that are separated from the last m_2 pickup points be at least one delivery point. Then the sequence can be denoted as P_i, $i = 1,\ldots,m_1, m_1+1,\ldots,m_1+m_2$ (i.e. $P_{m_1+m_2}$ is the last point).

We say that a packing plan for pickup point $P_{m_1+m_2}$ of a SOPP satisfies the cumulative request sequence constraint for loading sites (CRS-l) if the following conditions hold: (i) there are no boxes of a request j (loaded at pickup point P_j) between a box of request i and the rear of the vehicle; (ii) there are no boxes of request j above a box of request i ($i, j = 1,\ldots,m_1+m_2$, $j < i$). As shown in Männel and Bortfeldt (2015) the following proposition holds.

Proposition 1: Let a feasible plan for pickup point $P_{m_1+m_2}$ of a SOPP exist that meets the constraints (C1-l) and (C5) to (C7) and observes the CRS-l constraint. Then feasible packing plans observing constraints (C1-l) and (C5) to (C7) do also exist for pickup points P_i ($i = m_1+1,\ldots,m_1+m_2-1$).

In Männel and Bortfeldt (2015), a routing constraint was introduced in order to be able to derive feasible packing plans for the delivery points of a route. In the paper at hand the same purpose is achieved by additional packing constraints.

Let D_i be the corresponding delivery points of the pickup points P_i, $i = 1,\ldots,m_1+m_2$, of a SOPP. We say that a packing plan for pickup point $P_{m_1+m_2}$ satisfies the cumulative request sequence constraint for unloading sites (CRS-u1) if the following conditions hold: (i) if D_i lies before D_j, the boxes of request j must not lie between a box of request i and the rear of the vehicle ($i,j = 1,\ldots,m_1+m_2$); (ii) under the same assumptions, boxes of request j must not lie above a box of request i ($i,j = 1,\ldots,m_1+m_2$). The constraint (CRS-u2) is defined similarly, but only the second condition (ii) is required.

Proposition 2': Let a SOPP and a packing plan for the last pickup point $P_{m_1+m_2}$ be given.

(i) If the packing plan is feasible, meets the constraints (C1-l), (C5) to (C7) and satisfies the CRS-u1 constraint, then feasible packing plans, observing the constraints (C1-u) and (C5) to (C7), do exist for the consecutive m_3 delivery points behind $P_{m_1+m_2}$ ($m_3 \geq 1$).

(ii) If constraint CRS-u2 is substituted for CRS-u1, then feasible packing plans, observing constraints (C5) to (C7), do exist for the consecutive m_3 delivery points behind $P_{m_1+m_2}$ ($m_3 \geq 1$).

Proof: (i) Due to constraint CRS-u1, the boxes for the first delivery point behind $P_{m_1+m_2}$, say D_{i_1}, are in unloading position in the packing plan for $P_{m_1+m_2}$. If the boxes for D_{i_1} are removed, a packing plan for D_{i_1} results. Since the plan for $P_{m_1+m_2}$ observes support constraint (C6) and the removed boxes do not support boxes of other requests, the plan for D_{i_1} also meets (C6). Constraints (C5) and (C7) as well as feasibility conditions (FP1) to (FP3) hold, as they were met in plan $P_{m_1+m_2}$. The boxes for the next delivery point D_{i_2} are in unloading position due to constraint CRS-u1, i.e. the plan for D_{i_1} also meets constraint (C1-u). For the following delivery points, packing plans can be derived in a similar manner. (ii) Feasible packing plans, observing constraints (C5) to (C7), for consecutive delivery points behind $P_{m_1+m_2}$ can be derived as before. In particular, support constraint (C6) holds for these plans due to CRS-u2. □

Note that the constraints CRS-uj (j = 1,2) are formulated for all delivery points that correspond to pickup points P_i, $i = 1,\ldots,m_1+m_2$, of a given SOPP. However, proposition 2' only claims that feasible packing plans can be derived for the consecutive delivery points behind $P_{m_1+m_2}$.

In 3L-PDP variants 3 and 4, the reloading ban (C2) is required. It forbids that different placements of same boxes in packing plans for different sites of a route occur. We can state that if the reloading ban holds for all packing plans for the last points of sequences of open pickup points, then it holds for the derived packing plans for all other pickup and delivery points, too. Therefore, the above results show that for 3L-PDP variants 3 and 4 it is sufficient to construct feasible packing plans for the last pickup points of all sequences of open pickup points in a given route that meet the reloading ban. Feasible packing plans that observe the RS constraint (C1-l) and (C1-u) (in case of variant 4) and constraints (C5) to (C7) as well as the reloading ban can then be derived for all other pickup points and all delivery points of this route. Of course, this claim holds only if – for variant 3 – the constraints CRS-l and CRS-u2 are met in the plans for the last pickup points of SOPPs; for variant 4, the constraints CRS-l and CRS-u1 must be observed in these plans.

The procedure of packing checks for a route in 3L-PDP variants 3 and 4 is illustrated by an example in Figure 4.

Example for problem variant 3 and 4:

Given route

0 → P1 → P2 → D2 → P3 → P4 → P5 → D5 → D4 → P6 → D3 → D6 → D1 → 0

Sequence of open pickup points	Packing plan incl. constraints CRS-l CRS-uj (j = 1,2) to be provided for site	Derived packing plans result for sites
1. P1 → P2	P2	P1, D2
2. P1 → P3 → P4 → P5	P5	P3, P4, D5, D6
3. P1 → P3 → P6	P6	D3, D6, D1

- In 3L-PDP variant 3 constraint CRS-u2 has to be met at plans for sites P2, P5 and P6.
- In 3L-PDP variant 4 constraint CRS-u1 has to be met at plans for sites P2, P5 and P6.
- In both variants the packing plans to be provided for P2, P5 and P6 must include identical placements for boxes of
  - request 1 (occurring in plans for P2, P5 and P6)
  - request 3 (occurring in plans for P5 und P6).

Legend: 0: Depot, Pi / Di: pickup / delivery point of request i, i = 1,...,6.

Figure 4: Packing checks for 3L-PDP variants 3 and 4.

In 3L-PDP variant 4, there is no reloading effort at all, while in variant 3 some reloading effort can occur at delivery sites. If a vehicle arrives at a delivery site, all boxes of the corresponding request, say A, are to be unloaded. Since RS constraint (C1-u) is not required, some boxes of requests B, C, etc. may stand in the way of the A-boxes. These are called blocking boxes. We assume that blocking boxes have to be temporarily unloaded (and blocking boxes of blocking boxes, etc.). Because of constraint CRS-u2, blocking boxes cannot occur above boxes to be unloaded. For this reason, temporarily unloaded boxes can afterwards be loaded again so that they take their original placements.

There is no significant difference between insertion packing checks and acceptance packing checks, i.e. in any case for a given route the necessary feasible packing plans for last pickup points of SOPPs are to be provided.

4.4 Packing procedure

The packing procedure should be able to implement the reloading ban (C2). Packing plans to be generated for the last pickup points of SOPPs in a route need to be interrelated, i.e. if boxes are stowed in more than one of these packing plans, their placements must coincide. To ensure this the packing plans for a route are generated at once, i.e. by means of one and the same depth first search.

For the depth first search, a route is organized in multiple pickup and delivery sequences (PDS). A PDS contains the last m_2 ($m_2 \geq 1$) consecutive pickup points of a SOPP and the following m_3 ($m_3 \geq 1$) consecutive delivery points. A route consists of several PDSs and a packing plan is needed for each of these PDS, i.e. for its last pickup point.

The depth first search is carried out by means of the recursive procedure extend_packing_plan (see Figure 5) and the subordinated procedure initialize_packing_state (see Figure 6). The PDSs are indexed by *ipds*, the set *freeBoxes* includes the boxes of a PDS that are still available; *ipds* is set to zero and *freeBoxes* is set empty before the first call of the recursive procedure. The set *potentialPlacements* comprises potential placements of boxes in *freeBoxes*. Implemented placements for the current PDS are collected in the set *PDSPlan*, while the complete solution with the placements of all PDSs are held in the set *totalPlan*.

In procedure extend_packing_plan it is checked first whether the set *freeBoxes* is empty, i.e. whether the packing plan for the current PDS is complete. In this case (and if ipds > 0) this plan is incorporated in the complete solution *totalPlan*. The placements of boxes to be unloaded at delivery sites of the current PDS are marked in *totalPlan*.

```
extend_packing_plan (inout: ipds, freeBoxes, potentialPlacements, PDSPlan, totalPlan)
if number of procedure calls > maxEppCalls then abort packing check endif
if freeBoxes = ∅ then
        if ipds > 0 then
                totalPlan := totalPlan ∪ PDSPlan
                mark all placements in totalPlan as unloaded whose boxes belong
                to requests with delivery sites in PDS(ipds)
        endif
        ipds := ipds + 1                                                // next PDS
        if totalPlan complete then abort packing check endif
        initialize_packing_state(ipds, freeBoxes, potentialPlacements, PDSPlan, totalPlan)
endif
if there is at least one box in freeBoxes(ipds) without placement in potentialPlacements then return endif
provide list currentPlacements with potential placements that are currently to be tried
for i := 1 to |currentPlacements| do
        PDSPlan' :=  PDSPlan ∪ { currentPlacements(i) }                 // add placement to PDS-plan
        freeBoxes' := freeBoxes \ { currentPlacements(i).box }          // update free boxes
        potentialPlacements' := update(potentialPlacements)             // update potential placements
        extend_packing_plan (ipds, PDSPlan', freeBoxes', potentialPlacements', totalPlan) // recursive call
endfor
end.
```

Figure 5: Packing procedure 1: extend_packing_plan.

Afterwards index *ipds* is incremented and procedure initialize_packing_state is called for the new PDS. The complete solution *totalPlan* is only initialized empty for *ipds* = 1. The set *freeBoxes* is reinitialized and then includes the boxes that belong to the PDS. Potential placements for the whole set of boxes of current PDS in the lower left front corner of the loading space $L \times W \times H$ (cf. Figure 1) are generated.

```
initialize_packing_state (in: ipds, out: freeBoxes, potentialPlacements, PDSPlan, inout: totalPlan)
if ipds = 1 then totalPlan = ∅ endif
freeBoxes := { boxes to be loaded in PDS ipds }
initialize set potentialPlacements for box set freeBoxes and empty loading space
PDSPlan := ∅
for all placements Pl in totalPlan in given loading order not marked as unloaded do
        PDSPlan := PDSPlan ∪ { Pl }
        potentialPlacements  := update(potentialPlacements)
endfor
end.
```

Figure 6: Packing procedure 2: initialize_packing_state.

Then all placements, already put in *totalPlan* and *not* marked as unloaded, are reinserted in the new PDS solution *PDSPlan*. Each time another "old" placement is reinserted, the set *potentialPlacements* is updated taking into account all already inserted placements. After the *for*-loop is executed the current solution *PDSPlan* is filled with all placements of former PDSs that remain placements of the present PDS. As these placements are copied it is ensured that placements of same boxes in different PDS coincide. At the same time the set *potentialPlacements* at the end comprises only such placements which are compatible with all these "old" placements.

The current instance of procedure extend_packing_plan is aborted if there is at least one free box without a potential placement, i.e. if a complete solution can no longer be achieved on this search path. Candidates for the next placement for PDS *ipds* are selected from list *potentialPlacements* and provided in the list *currentPlacements*. All these placements are then tried alternatively. For each placement, the current PDS solution, the set of free boxes and the set of potential placements are updated accordingly, before procedure extend_packing_plan is called again. To update the list *potentialPlacements,* all potential placements are removed that can no longer be implemented. Additional potential reference points for new potential placements are determined as extreme points (see Crainic et al., 2008).

The selection of placements currently to be tried among all potential placements is governed by two rules. On the one hand, it is ensured that a vehicle is loaded from the front to the back, from bottom to top with lower priority, and from left to right with lowest priority. Hence, placements with smaller x-coordinates of the reference corner are preferred, etc. On the other hand, placements of boxes are preferred that belong to earlier loaded requests and, therefore, have to be stowed nearer to the cabin. The placement selection is controlled by the integer parameters *maxBoxRankDiff* and *maxRefPoints* where higher parameter values lead to a larger set of currently tried placements.

Potential placements are generated and updated in such a way that the packing plan for a PDS (i.e. for the last pickup point of the corresponding SOPP) is feasible and observes constraints (C5) to (C7), CRS-l and CRS-u1 (variant 4) or CRS-u2 (variant 3).

The packing check for a route is terminated when the number of recursive calls of procedure extend_packing_plan exceeds the specified limit *maxEppCalls* or a complete solution containing placements for all PDS is reached.

A cache of tested request sequences (routes) is used to accelerate the search as described in Männel and Bortfeldt (2015).

5 Computational experiments

In the computational experiments we test the hybrid algorithm for both “new” 3L-PDP variants 3 and 4 and the “old” variants 1A, 1B, 2 and 5 by means of 54 3L-PDP instances with up to 100 requests and up to 300 boxes. Moreover, we will hybridize the algorithm variants 3 and 5 as well as 4 and 5 in order to reduce the necessary packing effort and to generate high quality solutions quicker (see below).

The packing procedure is coded in the C++ programming language using Visual Studio 2012 Express, while the LNS scheme is implemented using the Java programming language under Eclipse 3.5.2. Preliminary experiments (in which total run times were varied) demonstrated that the impact of the different developing environments is negligible. All the experiments have been conducted on a PC with Intel Core i5-2500K (4.0 GHz, 16 GB RAM).

The generation of the benchmark instances is described completely in Männel and Bortfeldt (2015). Here we give a short overview of these instances and specify the parameter setting of the hybrid algorithm before the computational results are presented and analyzed.

5.1 Overview of benchmark instances for 3L-PDP

The 54 3L-PDP benchmark instances are overviewed in Table 3 with regard to number of requests, average number of boxes per request and distribution of pickup and delivery sites. The figures in columns 2−8 are instance numbers.

Table 3: Overview of the 54 3L-PDP benchmark instances.

Number of requests	2 Boxes per request on average			3 Boxes per request on average			Total
	Random	Mixed cluster	Pure cluster	Random	Mixed cluster	Pure cluster	
50	5	5	5	5	5	5	30
75	3	3	3	3	3	3	18
100	1	1	1	1	1	1	6

We distinguish the three distribution variants "Random", "Mixed cluster" and "Pure cluster". In variant

"Random", the sites are uniformly distributed in the plane, while they are clustered in the other variants. In variant "Mixed cluster" individual clusters may contain pickup as well as delivery sites, while only sites of one sort can occur in an individual cluster of variant "Pure clusters". Box dimensions are drawn randomly from the intervals [0.2·*L*, 0.6·*L*], [0.2·*W*, 0.6·*W*] and [0.2·*H*, 0.6·*H*], where *L*, *W* and *H* are the dimensions of the loading space (see Gendreau et al., 2006). A box is characterized as fragile with the probability 0.25. The percentage *a* for the minimal supporting area was specified as 0.75.

5.2 Parameter setting

The parameter setting for the experiments is specified in Table 4 and 5. The same parameterization of the routing procedure is used for all problem variants. All parameter values were determined based on limited computational experiments using a trial and error strategy.

Table 4: Parameter setting for the LNS routing procedure.

Parameter	Description	Value
r_{min}	lower bound of no. of removed customers	$0.04 \cdot n$
r_{max}	upper bound of no. of removed customers	$0.4 \cdot n$
W	start temperature control parameter	0.005
C	rate of geometrical cooling	0.9999
$p(Rh_R)$, $p(Rh_S)$	probability of Random / Shaw removal	0.3, 0.4
$p(Rh_W)$, $p(Rh_T)$	probability of Worst / Tour removal	0.1, 0.2
$p(Ih_G)$, $p(Ih_{R2})$, $p(Ih_{R3})$	probability of Greedy / Regret-2 / Regret-3 insert	0.1, 0.6, 0.3
w_{r1}, w_{r2}	weights of relatedness formula for Shaw removal	9, 2

Table 5: Parameter setting for packing procedure.

Parameter	Description	Value
maxEppCalls	Max. no. of calls of procedure extend_packing_plan	3000 / 200
maxBoxRankDiff	Max. tolerated rank difference of boxes	2
maxRefPoints	Max. number of admitted reference points	3

For parameter *maxEppCalls* we use the value 3000 if the checked route matches the IPR constraint (C3); otherwise the parameter is set to 200. In Table 6 the maximum run time per instance and single run is shown. The computing time depends on the number of requests and the average box number per request.

Table 6: Computing time limits in minutes for experiments

Number of requests	2 Boxes per request on avg.	3 Boxes per request on avg.
50	5	10
75	10	20
100	20	40

5.3 Computational results

Detailed results for the 54 3L-PDP instances regarding total travel distance (*ttd*) are presented in Table 7. In the leftmost column the instance names are listed. The next column shows the total travel distances for 3L-PDP (or algorithm) variant 5 where all constraints including the *request sequence constraint* for both loading and unloading sites (C1), the *reloading ban* (C2) and the *independent partial routes constraint* (C3) are considered. In the following eight columns total travel distances and gaps are indicated for the 3L-PDP variants 4, 4*, 3 and 3* (see Table 1).

In variant 4, the IPR constraint (C3) is not considered, while in variant 3 both IPR constraint (C3) and the RS constraint for unloading sites (C1-u) are not required. In the additional algorithm variants 3* and 4*, the variants 3 and 4, respectively, are hybridized with variant 5: in the first 40% of the computing time the algorithm has to construct routes which respect the IPR constraint (C3), i.e. it behaves as variant 5. We do this because the effort for packing checks strongly depends on the form of the routes, i.e. the algorithm can make much more iterations in the same time if it is restricted to IPR-routes because they are much easier to check than Non-IPR-routes.

All presented total travel distances are mean values over five runs. The corresponding gaps are calculated as (*ttd* - *ttd-V5*) / *ttd-V5* * 100 (%) (V5 stands for variant 5). In the last line of Table 7 the gap values of the 3L-PDP variants are averaged over the 54 instances.

Table 7: Results (travel distances) for variants 3, 3*, 4, 4* and 5 of 3L-PDP

Instance	Variant 5	Variant 4		Variant 4*		Variant 3		Variant 3*	
	ttd	ttd	gap (%)	ttd	gap (%)	ttd	gap (%)	ttd	gap (%)
50_RAND_2_1	1739.94	1643.11	-5.56	1629.81	-6.33	1570.34	-9.75	1602.51	-7.90
50_RAND_2_2	1580.00	1502.74	-4.89	1492.72	-5.52	1482.56	-6.17	1492.95	-5.51
50_RAND_2_3	1651.17	1518.44	-8.04	1531.83	-7.23	1502.72	-8.99	1493.03	-9.58
50_RAND_2_4	1588.00	1548.86	-2.46	1513.03	-4.72	1500.93	-5.48	1490.75	-6.12
50_RAND_2_5	1593.99	1486.20	-6.76	1487.01	-6.71	1455.88	-8.66	1474.06	-7.52
50_CLUS_2_1	1119.67	1064.42	-4.93	1071.83	-4.27	1052.90	-5.96	1058.39	-5.47
50_CLUS_2_2	1109.66	1047.73	-5.58	1058.34	-4.62	1028.68	-7.30	1025.30	-7.60
50_CLUS_2_3	1150.92	1086.31	-5.61	1081.56	-6.03	1085.97	-5.64	1064.77	-7.49
50_CLUS_2_4	1273.97	1230.84	-3.39	1229.59	-3.48	1204.04	-5.49	1197.63	-5.99
50_CLUS_2_5	1375.05	1308.44	-4.84	1308.79	-4.82	1294.11	-5.89	1301.14	-5.37
50_CPCD_2_1	1365.02	1349.48	-1.14	1334.86	-2.21	1338.68	-1.93	1341.77	-1.70
50_CPCD_2_2	1257.88	1279.74	1.74	1240.62	-1.37	1243.14	-1.17	1223.19	-2.76
50_CPCD_2_3	1231.69	1204.63	-2.20	1189.50	-3.43	1195.57	-2.93	1169.30	-5.07
50_CPCD_2_4	1327.56	1325.33	-0.17	1314.68	-0.97	1315.62	-0.90	1295.95	-2.38
50_CPCD_2_5	1459.16	1451.52	-0.52	1445.63	-0.93	1443.92	-1.04	1421.76	-2.56
50_RAND_3_1	1722.85	1584.28	-8.04	1593.35	-7.52	1587.37	-7.86	1595.07	-7.42
50_RAND_3_2	1567.63	1472.34	-6.08	1485.24	-5.26	1429.04	-8.84	1453.46	-7.28
50_RAND_3_3	1647.33	1559.67	-5.32	1540.46	-6.49	1494.60	-9.27	1523.61	-7.51
50_RAND_3_4	1562.71	1516.06	-2.99	1512.29	-3.23	1476.41	-5.52	1489.47	-4.69
50_RAND_3_5	1590.03	1505.75	-5.30	1505.09	-5.34	1431.41	-9.98	1456.89	-8.37
50_CLUS_3_1	1051.36	1028.23	-2.20	1022.56	-2.74	1030.12	-2.02	1017.71	-3.20
50_CLUS_3_2	1095.49	1021.71	-6.73	1022.18	-6.69	993.10	-9.35	1004.09	-8.34
50_CLUS_3_3	1122.79	1064.69	-5.17	1068.06	-4.87	1063.32	-5.30	1045.24	-6.91
50_CLUS_3_4	1254.53	1208.94	-3.63	1208.53	-3.67	1202.90	-4.12	1198.14	-4.50
50_CLUS_3_5	1322.85	1303.01	-1.50	1293.86	-2.19	1302.63	-1.53	1295.38	-2.08
50_CPCD_3_1	1333.81	1345.18	0.85	1341.98	0.61	1325.41	-0.63	1317.44	-1.23
50_CPCD_3_2	1242.96	1274.57	2.54	1240.26	-0.22	1245.30	0.19	1236.64	-0.51
50_CPCD_3_3	1241.13	1229.93	-0.90	1199.38	-3.36	1211.10	-2.42	1198.64	-3.42
50_CPCD_3_4	1307.31	1318.83	0.88	1311.42	0.31	1306.57	-0.06	1291.73	-1.19
50_CPCD_3_5	1437.59	1440.46	0.20	1447.30	0.68	1428.73	-0.62	1438.87	0.09
75_RAND_2_1	2127.10	2096.55	-1.44	2062.46	-3.04	2077.60	-2.33	2039.35	-4.13
75_RAND_2_2	2130.19	2057.50	-3.41	2013.28	-5.49	2027.33	-4.83	1977.09	-7.19
75_RAND_2_3	2182.23	2099.81	-3.78	2106.39	-3.48	2028.58	-7.04	2004.88	-8.13
75_CLUS_2_1	1465.83	1439.45	-1.80	1410.28	-3.79	1426.21	-2.70	1383.24	-5.63
75_CLUS_2_2	1426.07	1395.63	-2.13	1370.41	-3.90	1385.26	-2.86	1351.14	-5.25
75_CLUS_2_3	1489.35	1481.52	-0.53	1448.61	-2.74	1446.28	-2.89	1423.25	-4.44
75_CPCD_2_1	2220.47	2191.00	-1.33	2188.05	-1.46	2163.60	-2.56	2139.78	-3.63
75_CPCD_2_2	2207.16	2252.08	2.04	2166.06	-1.86	2195.40	-0.53	2146.77	-2.74
75_CPCD_2_3	2278.44	2270.61	-0.34	2228.77	-2.18	2215.24	-2.77	2181.08	-4.27
75_RAND_3_1	2146.48	2125.46	-0.98	2079.43	-3.12	2067.85	-3.66	2086.61	-2.79
75_RAND_3_2	2068.24	2020.77	-2.30	2009.51	-2.84	1951.84	-5.63	1951.97	-5.62
75_RAND_3_3	2115.67	2053.03	-2.96	2051.64	-3.03	1993.45	-5.78	1999.77	-5.48
75_CLUS_3_1	1449.74	1453.46	0.26	1426.23	-1.62	1458.87	0.63	1439.73	-0.69
75_CLUS_3_2	1424.63	1405.01	-1.38	1388.71	-2.52	1389.04	-2.50	1383.26	-2.90
75_CLUS_3_3	1473.63	1485.06	0.78	1457.45	-1.10	1458.15	-1.05	1443.64	-2.03
75_CPCD_3_1	2229.25	2237.87	0.39	2211.88	-0.78	2200.71	-1.28	2173.53	-2.50
75_CPCD_3_2	2166.01	2251.96	3.97	2239.99	3.42	2197.36	1.45	2192.49	1.22
75_CPCD_3_3	2222.43	2224.28	0.08	2220.07	-0.11	2204.27	-0.82	2208.14	-0.64
100_RAND_2_1	4088.54	4077.67	-0.27	3970.83	-2.88	3967.35	-2.96	3912.13	-4.31
100_CLUS_2_1	4197.41	4141.66	-1.33	4105.80	-2.18	4151.53	-1.09	4046.64	-3.59
100_CPCD_2_1	4274.19	4347.95	1.73	4363.07	2.08	4364.80	2.12	4270.70	-0.08
100_RAND_3_1	4014.22	4022.28	0.20	3978.07	-0.90	3939.48	-1.86	3890.32	-3.09
100_CLUS_3_1	4102.11	4202.41	2.45	4149.09	1.15	4127.51	0.62	4076.94	-0.61
100_CPCD_3_1	4166.71	4347.99	4.35	4240.65	1.77	4379.58	5.11	4197.89	0.75
Average gap			**-1.95**		**-2.84**		**-3.52**		**-4.21**

By algorithm variant 4 (4*) a mean reduction of total travel distance by 1.95% (2.84%) is achieved compared to variant 5. The corresponding reduction reached by variant 3 (3*) amounts to 3.52% (4.21%). However, there is no reloading effort with variant 4 (4*), while in variant 3 (3*) the improvement in terms of travel distance is „bought“ by a portion of reloading effort. The hybridized algorithm variants 3* and 4*, in which the search is temporarily restricted to IPR-routes, turn out to be rather successful and achieve improvements of 0.7 %-points (3* vs. 3) and 0.9 %-points (4* vs. 4).

Tables 8 and 9 indicate the influence of instance size and type (regarding distribution of sites) on the solution quality for different algorithm variants. For small instances with up to 50 requests the variants 3 and 4 achieve significant better results than variant 5, while for large instances with 100 requests variant 5 performs better than 3 and 4. However, variants 3* and 4*, which temporarily restrict the search space, show their strength just for large and difficult instances and perform even better than variant 5. On the other hand, the difference between variant 3* (4*) and variant 3(4) is almost negligible for small instances.

With regard to the instance types „Random“, „Mixed cluster“ and „Pure cluster“ it can be observed that variants 3 and 4 yield largest improvements compared with variant 5 for instance type „Random“ and provided smallest improvements (or even worsening) for type „Pure cluster“. Again, a significant improvement of results was achieved by the hybridized variants 3* and 4* and it was reached especially for the „problematic“ instance type „Pure cluster“.

Table 8: Average gap for small, midsize and large instances

Number of requests	Variant 4 Average gap in %	Variant 4* Average gap in %	Variant 3 Average gap in %	Variant 3* Average gap in %
50	-3.26	-3.75	-4.82	-4.99
75	-0.83	-2.20	-2.62	-3.71
100	1.19	-0.16	0.32	-1.82

Table 9: Average gap for “random”, “mixed cluster” and “pure cluster” instances

Type	Variant 4 Average gap in %	Variant 4* Average gap in %	Variant 3 Average gap in %	Variant 3* Average gap in %
Random	-3.91	-4.62	-6.37	-6.26
Mixed cluster	-2.63	-3.34	-3.58	-4.56
Pure cluster	0.68	-0.56	-0.60	-1.81

In the following we deal with the tradeoff between travel distance and reloading quantity. In variants 4, 4*, and 5 there is no reloading effort as the *Reloading ban* (C2) is in force (see Table 1). Among the variants with Reloading ban variant 4* provides the best results in terms of total travel distance. Thus, variant 4* will be compared now with 3L-PDP algorithm variants 1A, 1B, 2 and 3* regarding total travel distance and reloading effort (for simplification variant 3 is omitted here).

Table 10 is organized as Table 7 and shows the total travel distances and gaps (as percentages) based on variant 4*.

The reloading effort needed for a 3L-PDP instance is primarily given as reloading quantity, i.e. as the weight of all boxes that are reloaded. If a box is reloaded, say, two times the weight of the box is counted two times. Thus it may occur that the reloading quantity exceeds the total weight of the boxes. Table 11 is organized as follows. The first column includes the instance names and the second column shows the total weight of all requests per instance (cargo weight). In the following eight columns the reloading quantities for the relevant 3L-PDP variants are given as absolute values (in weight units) and as percentages of the cargo weight. The results per instance are, again, averaged over five runs. In the last line of Table 11 the percentaged reloading quantities are averaged over the 54 instances. Since the reloading effort is zero for problem variant 4*, this variant does not occur in Table 11.

The reloading quantities of variant 2 (missing Reloading ban) and 3* (missing RS constraint for unloading sites) are moderate and amount to 15.84% and 24.59% of the cargo weight on average. For problem variants 1A and 1B, where both constraints are missing, the mean reloading quantity is much higher (101.09% and 90.94%, respectively). However, the variants 2 and 3* bring only a small decrease of the total travel distance (0.88% and 1.41%) while the variants 1A and 1B reduce the total travel distance much stronger (9.47% and 9.15%). Table 12 summarizes the results regarding total travel distance and reloading effort. For each 3L-PDP variant the total travel distance is now given as percentage of the travel distance of variant 4* while the reloading quantities are again indicated as percentages of the cargo weight. All presented values are averaged over the five runs per instance and over the 54 3L-PDP instances.

The results for variants 1A, 1B, 2 and 3* show the tradeoff between travel distances and reloading effort indicating that a saving of travel distance has to be "paid" with an additional portion of reloading effort. The indicated figures for the 3L-PDP variants correspond very well with the expected differences between those variants regarding travel distances and reloading effort as shown in Table 1.

Table 10: Results (travel distances) for variants 1A, 1B, 2, 3* and 4* of 3L-PDP

Instance	Variant 4*	Variant 3*		Variant 2		Variant 1A		Variant 1B	
	ttd	ttd	gap (%)	ttd	gap (%)	ttd	gap (%)	ttd	gap (%)
50_RAND_2_1	1629.81	1602.51	-1.68	1637.09	0.45	1444.45	-11.37	1446.36	-11.26
50_RAND_2_2	1492.72	1492.95	0.02	1516.85	1.62	1311.35	-12.15	1320.46	-11.54
50_RAND_2_3	1531.83	1493.03	-2.53	1563.20	2.05	1321.50	-13.73	1329.82	-13.19
50_RAND_2_4	1513.03	1490.75	-1.47	1541.67	1.89	1358.01	-10.25	1345.53	-11.07
50_RAND_2_5	1487.01	1474.06	-0.87	1541.45	3.66	1335.51	-10.19	1344.54	-9.58
50_CLUS_2_1	1071.83	1058.39	-1.25	1038.29	-3.13	973.46	-9.18	977.61	-8.79
50_CLUS_2_2	1058.34	1025.30	-3.12	1038.32	-1.89	922.93	-12.79	925.57	-12.54
50_CLUS_2_3	1081.56	1064.77	-1.55	1097.44	1.47	983.77	-9.04	999.44	-7.59
50_CLUS_2_4	1229.59	1197.63	-2.60	1217.47	-0.99	1112.42	-9.53	1111.25	-9.62
50_CLUS_2_5	1308.79	1301.14	-0.58	1299.86	-0.68	1221.03	-6.70	1227.57	-6.21
50_CPCD_2_1	1334.86	1341.77	0.52	1300.55	-2.57	1245.24	-6.71	1248.23	-6.49
50_CPCD_2_2	1240.62	1223.19	-1.40	1226.18	-1.16	1158.42	-6.63	1150.53	-7.26
50_CPCD_2_3	1189.50	1169.30	-1.70	1178.45	-0.93	1102.03	-7.35	1117.55	-6.05
50_CPCD_2_4	1314.68	1295.95	-1.42	1306.18	-0.65	1235.71	-6.01	1244.43	-5.34
50_CPCD_2_5	1445.63	1421.76	-1.65	1433.12	-0.87	1361.01	-5.85	1375.63	-4.84
50_RAND_3_1	1593.35	1595.07	0.11	1610.28	1.06	1439.54	-9.65	1441.52	-9.53
50_RAND_3_2	1485.24	1453.46	-2.14	1456.14	-1.96	1282.98	-13.62	1283.29	-13.60
50_RAND_3_3	1540.46	1523.61	-1.09	1560.56	1.30	1308.44	-15.06	1331.61	-13.56
50_RAND_3_4	1512.29	1489.47	-1.51	1541.03	1.90	1335.19	-11.71	1323.85	-12.46
50_RAND_3_5	1505.09	1456.89	-3.20	1536.97	2.12	1335.95	-11.24	1340.18	-10.96
50_CLUS_3_1	1022.56	1017.71	-0.47	1006.26	-1.59	958.17	-6.30	960.20	-6.10
50_CLUS_3_2	1022.18	1004.09	-1.77	1023.18	0.10	904.41	-11.52	914.41	-10.54
50_CLUS_3_3	1068.06	1045.24	-2.14	1074.95	0.65	973.62	-8.84	988.03	-7.49
50_CLUS_3_4	1208.53	1198.14	-0.86	1195.44	-1.08	1087.79	-9.99	1097.94	-9.15
50_CLUS_3_5	1293.86	1295.38	0.12	1279.98	-1.07	1225.50	-5.28	1225.62	-5.27
50_CPCD_3_1	1341.98	1317.44	-1.83	1311.53	-2.27	1260.72	-6.06	1247.57	-7.03
50_CPCD_3_2	1240.26	1236.64	-0.29	1215.54	-1.99	1173.01	-5.42	1190.41	-4.02
50_CPCD_3_3	1199.38	1198.64	-0.06	1200.31	0.08	1101.86	-8.13	1117.87	-6.80
50_CPCD_3_4	1311.42	1291.73	-1.50	1286.49	-1.90	1250.68	-4.63	1262.89	-3.70
50_CPCD_3_5	1447.30	1438.87	-0.58	1412.09	-2.43	1371.73	-5.22	1371.39	-5.25
75_RAND_2_1	2062.46	2039.35	-1.12	2038.21	-1.18	1840.43	-10.77	1821.77	-11.67
75_RAND_2_2	2013.28	1977.09	-1.80	2030.27	0.84	1731.18	-14.01	1744.05	-13.37
75_RAND_2_3	2106.39	2004.88	-4.82	2086.63	-0.94	1796.50	-14.71	1830.70	-13.09
75_CLUS_2_1	1410.28	1383.24	-1.92	1390.68	-1.39	1305.25	-7.45	1302.66	-7.63
75_CLUS_2_2	1370.41	1351.14	-1.41	1378.87	0.62	1250.74	-8.73	1256.70	-8.30
75_CLUS_2_3	1448.61	1423.25	-1.75	1429.92	-1.29	1321.30	-8.79	1323.52	-8.64
75_CPCD_2_1	2188.05	2139.78	-2.21	2153.75	-1.57	1991.91	-8.96	1978.99	-9.55
75_CPCD_2_2	2166.06	2146.77	-0.89	2162.77	-0.15	2012.80	-7.08	2015.98	-6.93
75_CPCD_2_3	2228.77	2181.08	-2.14	2201.15	-1.24	2072.41	-7.02	2077.58	-6.78
75_RAND_3_1	2079.43	2086.61	0.35	2038.48	-1.97	1854.33	-10.83	1857.94	-10.65
75_RAND_3_2	2009.51	1951.97	-2.86	1950.25	-2.95	1683.86	-16.21	1687.59	-16.02
75_RAND_3_3	2051.64	1999.77	-2.53	2025.61	-1.27	1765.98	-13.92	1764.03	-14.02
75_CLUS_3_1	1426.23	1439.73	0.95	1393.02	-2.33	1299.61	-8.88	1321.71	-7.33
75_CLUS_3_2	1388.71	1383.26	-0.39	1378.46	-0.74	1238.43	-10.82	1243.10	-10.49
75_CLUS_3_3	1457.45	1443.64	-0.95	1412.18	-3.11	1308.45	-10.22	1319.44	-9.47
75_CPCD_3_1	2211.88	2173.53	-1.73	2162.30	-2.24	2058.18	-6.95	2012.76	-9.00
75_CPCD_3_2	2239.99	2192.49	-2.12	2169.31	-3.16	2040.09	-8.92	2050.58	-8.46
75_CPCD_3_3	2220.07	2208.14	-0.54	2185.07	-1.58	2089.08	-5.90	2090.38	-5.84
100_RAND_2_1	3970.83	3912.13	-1.48	3996.65	0.65	3449.25	-13.14	3435.59	-13.48
100_CLUS_2_1	4105.80	4046.64	-1.44	4006.46	-2.42	3593.84	-12.47	3645.81	-11.20
100_CPCD_2_1	4363.07	4270.70	-2.12	4195.00	-3.85	4125.16	-5.45	4117.60	-5.63
100_RAND_3_1	3978.07	3890.32	-2.21	3938.09	-1.00	3475.00	-12.65	3469.34	-12.79
100_CLUS_3_1	4149.09	4076.94	-1.74	3935.40	-5.15	3613.59	-12.91	3634.02	-12.41
100_CPCD_3_1	4240.65	4197.89	-1.01	4196.85	-1.03	4063.02	-4.19	4053.98	-4.40
Average gap			**-1.41**		**-0.88**		**-9.47**		**-9.15**

Table 11: Results (reloading quantities) for variants 1A, 1B, 2, 3* of 3L-PDP

Instance	Cargo weight	Variant 3* reloading quantity absolute	in %	Variant 2 reloading quantity absolute	in %	Variant 1A reloading quantity absolute	in %	Variant 1B reloading quantity Abso-lute	in %
50_RAND_2_1	610544	139947	22.92	168844	27.65	575543	94.27	481167	78.81
50_RAND_2_2	578322	111520	19.28	144603	25.00	611581	105.75	546683	94.53
50_RAND_2_3	530415	114087	21.51	134218	25.30	656712	123.81	605315	114.12
50_RAND_2_4	652932	164478	25.19	75086	11.50	617234	94.53	555832	85.13
50_RAND_2_5	698040	65187	9.34	147825	21.18	630059	90.26	570085	81.67
50_CLUS_2_1	610544	135750	22.23	75706	12.40	497145	81.43	488510	80.01
50_CLUS_2_2	578322	94858	16.40	73193	12.66	591082	102.21	538623	93.14
50_CLUS_2_3	530415	137406	25.91	97984	18.47	615979	116.13	555255	104.68
50_CLUS_2_4	652932	159141	24.37	113269	17.35	628491	96.26	627601	96.12
50_CLUS_2_5	698040	101890	14.60	63178	9.05	441152	63.20	445383	63.80
50_CPCD_2_1	610544	179234	29.36	38459	6.30	531662	87.08	444986	72.88
50_CPCD_2_2	578322	178001	30.78	45819	7.92	540811	93.51	501153	86.66
50_CPCD_2_3	530415	106848	20.14	20312	3.83	475366	89.62	443035	83.53
50_CPCD_2_4	652932	141631	21.69	34114	5.22	449425	68.83	386894	59.25
50_CPCD_2_5	698040	141822	20.32	39561	5.67	465366	66.67	395123	56.60
50_RAND_3_1	611295	140306	22.95	192314	31.46	478161	78.22	456339	74.65
50_RAND_3_2	579037	199194	34.40	161082	27.82	605642	104.59	514886	88.92
50_RAND_3_3	531236	194096	36.54	94288	17.75	679188	127.85	574234	108.09
50_RAND_3_4	654049	166122	25.40	28006	4.28	626539	95.79	535115	81.82
50_RAND_3_5	699080	151823	21.72	65109	9.31	517152	73.98	495000	70.81
50_CLUS_3_1	611295	103551	16.94	164188	26.86	552782	90.43	516636	84.52
50_CLUS_3_2	579037	162539	28.07	69990	12.09	505169	87.24	499791	86.31
50_CLUS_3_3	531236	155035	29.18	115331	21.71	591276	111.30	537444	101.17
50_CLUS_3_4	654049	70761	10.82	59088	9.03	625672	95.66	582650	89.08
50_CLUS_3_5	699080	132591	18.97	76706	10.97	521244	74.56	495218	70.84
50_CPCD_3_1	611295	132567	21.69	31358	5.13	483517	79.10	449812	73.58
50_CPCD_3_2	579037	169281	29.23	53327	9.21	647999	111.91	492084	84.98
50_CPCD_3_3	531236	160991	30.30	57255	10.78	479460	90.25	410820	77.33
50_CPCD_3_4	654049	153566	23.48	39817	6.09	470149	71.88	391110	59.80
50_CPCD_3_5	699080	169068	24.18	35327	5.05	489315	69.99	403293	57.69
75_RAND_2_1	772435	168053	21.76	196123	25.39	948928	122.85	813012	105.25
75_RAND_2_2	780361	193887	24.85	121514	15.57	960353	123.07	834959	107.00
75_RAND_2_3	808203	281873	34.88	103054	12.75	940073	116.32	852795	105.52
75_CLUS_2_1	772435	174153	22.55	150562	19.49	905700	117.25	754982	97.74
75_CLUS_2_2	780361	180800	23.17	124061	15.90	890977	114.18	722868	92.63
75_CLUS_2_3	808203	190353	23.55	135203	16.73	1031982	127.69	917112	113.48
75_CPCD_2_1	772435	193212	25.01	117514	15.21	721007	93.34	729654	94.46
75_CPCD_2_2	780361	186023	23.84	105109	13.47	742203	95.11	652853	83.66
75_CPCD_2_3	808203	212359	26.28	57111	7.07	726025	89.83	719133	88.98
75_RAND_3_1	774140	157765	20.38	153782	19.86	832513	107.54	889335	114.88
75_RAND_3_2	782381	250816	32.06	171963	21.98	1002422	128.12	828062	105.84
75_RAND_3_3	810106	221559	27.35	167784	20.71	956912	118.12	874143	107.90
75_CLUS_3_1	774140	227150	29.34	213879	27.63	954814	123.34	820840	106.03
75_CLUS_3_2	782381	191653	24.50	228154	29.16	876780	112.07	710735	90.84
75_CLUS_3_3	810106	183844	22.69	100649	12.42	972879	120.09	961490	118.69
75_CPCD_3_1	774140	257150	33.22	142512	18.41	840097	108.52	749999	96.88
75_CPCD_3_2	782381	236430	30.22	157019	20.07	805035	102.90	741657	94.79
75_CPCD_3_3	810106	206884	25.54	75782	9.35	788439	97.33	750777	92.68
100_RAND_2_1	1072407	276456	25.78	164455	15.34	1381179	128.79	1235513	115.21
100_CLUS_2_1	1072407	224647	20.95	267199	24.92	1312878	122.42	1165780	108.71
100_CPCD_2_1	1072407	225515	21.03	192813	17.98	1146020	106.86	1019087	95.03
100_RAND_3_1	1074809	377887	35.16	212719	19.79	1400979	130.35	1215913	113.13
100_CLUS_3_1	1074809	293609	27.32	263470	24.51	1301413	121.08	1138620	105.94
100_CPCD_3_1	1074809	309142	28.76	154385	14.36	1025408	95.40	1019520	94.86
Average			**24.59**		**15.84**		**101.09**		**90.94**

Table 12: Tradeoff between total travel distance and reloading quantity

3L-PDP variant	Total travel distance in %	Reloading quantity average in %
4*	100.00	0.00
3*	98.59	24.59
2	99.12	15.84
1A	90.53	101.09
1B	90.85	90.94

6 Conclusions

In the paper at hand and in the previous paper by Männel and Bortfeldt (2015), the vehicle routing problem with pickup and delivery (PDP) has been extended to an integrated vehicle routing and loading problem. In such problems 3D rectangular items have to be transported in homogeneous vehicles with a rectangular 3D loading space (3L-PDP). In the problem formulation we concentrated on the question under which conditions any reloading effort, i.e. any movement of boxes *after* loading and *before* unloading, can be avoided. It turned out that the request sequence constraint (C1) for loading and unloading sites is not sufficient. Instead, we must require either a new routing constraint, called independent partial routes condition (C3), or a new packing constraint, termed reloading ban (C2), to exclude any reloading effort. Eventually, a spectrum of five 3L-PDP variants was introduced that allow for different portions of reloading effort and reciprocal savings of travel distance.

In this paper, we focused on the so-called reloading ban, a packing constraint that ensures identical placements of same boxes in different packing plans. A hybrid algorithm for solving the 3L-PDP with reloading ban consisting of a routing and a packing procedure has been proposed. The routing procedure, adopted from Ropke and Pisinger (2006), performs a large neighborhood search. A tree search heuristic, originally published by Bortfeldt (2012), is responsible for packing boxes. To cope with the reloading ban, the packing procedure had to be substantially extended. The main point is that, given the reloading ban to be observed, a single 3L-PDP route requires multiple *interrelated* packing plans. That is, if boxes are stowed in more than one of these packing plans, their placements must coincide.

The hybrid algorithm (variant 4), proposed in the paper at hand, was tested by means of 54 3L-PDP instances with up to 100 requests and up to 300 boxes. A comparison was made with the hybrid algorithm which was proposed in Männel and Bortfeldt (2015) for the 3L-PDP with independent partial routes condition (variant 5). The new algorithm reaches noticeable smaller travel distances compared with the rival variant 5. This result seems plausible since in variant 5 any reloading effort is excluded only by special shapes of routes, i.e. the search space is strongly restricted. However, the difference of travel distance is only ca. 2% on average due to the higher computational burden of the new algorithm. Therefore, both variants were hybridized at last and this resulted in a further improvement of travel distances. A comparison with other variants of the hybrid algorithms shows a clear tradeoff between travel distance and reloading effort and confirms the theoretical expectations.

Future research regarding the 3L-PDP should consider further constraints, e.g. regarding load stability, which are indispensable requirements in practice.

References

Bartók, T; Imre, C (2011): Pickup and Delivery Vehicle Routing with Multidimensional Loading Constraints. *Acta Cybernetica*, 20, 17-33.

Bent, R; van Hentenryck, P (2006): A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33:875-893.

Bortfeldt, A (2012): A Hybrid Algorithm for the Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints. *Computers & Operations Research*, 39:2248-2257.

Bortfeldt, A; Homberger, J (2013): Packing First, Routing Second - a Heuristic for the Vehicle Routing and Loading Problem. *Computers & Operations Research*, 40:873-885.

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*, 243:82-96.

Crainic, TG; Perboli, G; Tadei, R (2008): Extreme Point-based Heuristics for Three-dimensional Bin Packing. *INFORMS Journal on Computing*, 20:368-384.

Fuellerer, G; Doerner, KF; Hartl, R; Iori, M (2010): Metaheuristics for Vehicle Routing Problems with Three-dimensional Loading Constraints. *European Journal of Operational Research*, 201:751-759.

Gendreau, M; Iori, M; Laporte, G; Martello, S (2006): A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transportation Science*, 40:342-350.

Li, H; Lim, A (2001): A metaheuristic for the pickup and delivery problem with time windows. 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01). IEEE Computer Society, Los Alamitos, CA, 333-340.

Männel, D; Bortfeldt, A: A Hybrid Algorithm for the Vehicle Routing Problem with Pickup and Delivery and 3D Loading Constraints. *Working Paper No. 15/2015*, Faculty of Economics and Management, Otto-von-Guericke University Magdeburg.

Malapert, A; Guéret, C; Jussien, N; Langevin, A; Rousseau, LM (2008): Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints. *Proceedings of the First CPAIOR Workshop on Bin Packing and Placement Constraints (BPPC'08)*, Paris, France.

Moura, A; Oliveira, JF (2009): An Integrated Approach to Vehicle Routing and Container Loading Problems. *Operations Research Spectrum* 31:775-800.

Parragh, SN; Doerner, KF; Hartl, RF (2008): A Survey on Pickup and Delivery Problems. Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58:81-117.

Pollaris, H; Braekers, K; Caris, A; Janssens, G; Limbourg, S (2015): Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37:297-330.

Ropke, S; Pisinger, D (2006): An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40:455-472.

Tao, Y; Wang, F (2015): An effective tabu search approach with improved loading algorithms for the 3L-CVRP. *Computers & Operations Research*, 55:127-140.

Tarantilis, CD; Zachariadis, EE; Kiranoudis, CT (2009): A Hybrid Metaheuristic Algorithm for the Integrated Vehicle Routing and Three-dimensional Container-loading Problem. *IEEE Transactions on Intelligent Transportation Systems,* 10:255-271.

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274