European Journal of Operational Research 282 (2020) 545–558

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints

Andreas Bortfeldt[a], Junmin Yi[b,*]

[a] Otto-von-Guericke-University Magdeburg, Germany
[b] Xiamen University of Technology, China

ARTICLE INFO

Article history:
Received 23 December 2018
Accepted 12 September 2019
Available online 26 September 2019

Keywords:
Routing
Packing
Split delivery
3D loading constraints
Hybrid algorithm

ABSTRACT

The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints (3L-SDVRP) combines vehicle routing and three-dimensional loading with additional packing constraints. In the 3L-SDVRP splitting deliveries of customers is basically possible, i.e. a customer can be visited in two or more tours. We examine essential problem features and introduce two problem variants. In the first variant, called 3L-SDVRP with *forced* splitting, a delivery is only split if the demand of a customer cannot be transported by a single vehicle. In the second variant, termed 3L-SDVRP with *optional* splitting, splitting customer deliveries can be done any number of times. We propose a hybrid algorithm consisting of a local search algorithm for routing and a genetic algorithm and several construction heuristics for packing. Numerical experiments are conducted using three sets of instances with both industrial and academic origins. One of them was provided by an automotive logistics company in Shanghai; in this case some customers per instance have a total freight volume larger than the loading space of a vehicle. The results prove that splitting deliveries can be beneficial not only in the one-dimensional case but also when goods are modeled as three-dimensional items.

1. Introduction

In the distribution of goods and other logistic operations vehicle routing problems (VRP) are of eminent importance. In the Capacitated VRP (CVRP) as well as in numerous derived VRPs that address more complicated situations, each customer has to be visited only once. Thus, the entire demand must be delivered by one vehicle during one visit. In the Split Delivery VRP (SDVRP) this constraint is relaxed and the demand of a customer can be delivered by two or more vehicles. Both numerical experiments with solution methods and theoretical results prove that large savings in terms of travel distance can be obtained if splitting deliveries is allowed (see, e.g., Archetti and Speranza, 2012).

In this paper, we study whether splitting deliveries remains beneficial, if the SDVRP is extended to the SDVRP with three-dimensional (3D) loading constraints (3L-SDVRP). The CVRP with three-dimensional loading constraints (3L-CVRP) was introduced by Gendreau, Iori, Laporte, and Martello (2006). Contrasting to the classical capacitated VRP, customer demands are given by sets of rectangular shaped pieces (boxes) and the scalar capacity of vehicles is changed to a 3D loading space. Moreover, packing constraints that occur frequently in general cargo transportation and are related to containers, items, cargo and load (Bortfeldt and Wäscher, 2013; Pollaris, Braekers, Caris, Janssens, and Limbourg, 2015) can be included. A solution consists of routes with corresponding packing plans where each packing plan stows the boxes of all customers visited in the same route. Extending "pure" routing problems to vehicle routing and loading problems with 3D loading constraints (3L-VRP) generally allows for a more realistic handling of real-word routing problems where general cargo is to be delivered or collected (Iori and Martello, 2010, 2013; Pollaris et al., 2015, Côté, Guastaroba, and Speranza, 2017).

The 3L-SDVRP has been treated very rarely in the literature so far. Ceschia, Schaerf, and Stützle (2013) proposed a local-search algorithm for solving the 3L-CVRP and 3L-SDVRP. While they achieved good results for the 3L-CVRP no distance savings could be obtained by permitting split deliveries.

In the following, we will examine essential features and variants of 3L-SDVRP, introduce a 3L-SDVRP formulation and propose a hybrid algorithm for solving the 3L-SDVRP. The algorithm should be able to generate high-quality solutions with a simple structure of packing plans within short running times. We study the 3L-SDVRP in the context of factory inbound logistics where the box sets of the customers are typically *weakly* heterogeneous. We apply our algorithm to three sets of instances. The data of the first

* Corresponding author.
E-mail addresses: andreas.bortfeldt@ovgu.de (A. Bortfeldt), yijunmin@xmut.edu.cn (J. Yi).

https://doi.org/10.1016/j.ejor.2019.09.024
0377-2217/

set stem from a Shanghai automotive logistics company. The second set of instances comes from Ceschia et al. (2013) with data originated from an Italian industrial partner. The third set is constructed using well-known benchmark instances for the CVRP and the container loading problem (CLP).

The paper is organized as follows. In Section 2, the relevant literature is reviewed and in Section 3, we identify essential features and variants of the 3L-SDVRP and formulate this problem with all details. In Section 4, we introduce a hybrid algorithm for the 3L-SDVRP. Section 5 is dedicated to the numerical experiments. Finally, in Section 6 we present concluding remarks and an outlook to future research.

2. Literature review

In this section, the literature concerning the SDVRP and 3L-VRPs is reviewed and existing approaches to the 3L-SDVRP are shortly described.

2.1. Vehicle routing problem with split deliveries

The SDVRP is defined, roughly spoken, as the classical CVRP except that splitting deliveries is allowed. Thus, a customer may be visited by several vehicles if beneficial. Moreover, customers can exist with large demands exceeding the vehicle capacity; obviously, this situation requires the possibility to split deliveries.

The SDVRP was introduced by Dror and Trudeau (1989, 1990) and extensively analyzed by, e.g., Archetti, Savelsbergh, and Speranza (2008) and Archetti, Bianchessi, and Speranza (2011). Interesting results relate for example to worst-case analyses in order to quantify how much worse a solution without split deliveries can be compared to SDVRP solutions. Exact approaches were proposed amongst others by Belenguer, Martinez, and Mota (2000), Dror, Laporte, and Trudeau (1994), and Moreno, Poggi de Aragão, and Uchoa (2010). The Branch-and-Cut-and-Price algorithm by Archetti, Bianchessi, and Speranza (2014) and the method proposed by Ozbaygin, Karasan, and Yaman (2018) based on a new vehicle-indexed flow formulation belong currently to the strongest exact methods. Since only small instances can be solved using exact methods so far, the SDVRP is mostly tackled by metaheuristic methods. These include local search (Dror and Trudeau, 1989, 1990), tabu search (TS, Archetti, Speranza, and Hertz, 2006, Berbotto, García, and Nogales, 2014, Qiu, Fu, Eglese, and Tang, 2018), genetic algorithm (GA, Wilck and Cavalier, 2012), hybrid metaheuristics (Rajappa, Wilck, and Bell, 2016) and particle swarm optimization (Shi, Zhang, Wang, and Fang, 2018). Recently, high-quality results have been obtained by the use of aggregated/disaggregated mathematical formulations, such as Benders decomposition, valid inequalities, and tailored optimization techniques based on branch-and-cut frameworks (Bruck and Iori, 2017), and a new and tailored branch-and-cut algorithm for SDVRPTW (Bianchessi and Irnich, 2019). For surveys of the SDVRP literature, the reader is referred to Archetti and Speranza (2012) and Irnich, Schneider, and Vigo (2014).

2.2. Vehicle routing problems with 3D loading constraints

Pollaris et al. (2015) survey the state of the art in the area of combined vehicle routing and loading problems (see also Iori and Martello, 2010).

The 3L-CVRP was introduced by Gendreau et al. (2006) and motivated by a real furniture distribution decision in Italy. Besides geometrical constraints, exact-one-visit condition and a weight constraint they considered four loading constraints on orientation, fragility, vertical stability and last-in-first-out policy (LIFO). The 3L-CVRP with these constraints is called here the *Gendreau formulation* of 3L-CVRP.

A number of heuristic approaches for solving the 3L-CVRP in Gendreau formulation have been proposed in recent years. Most of the solution methods are hybrids consisting of a metaheuristic algorithm for routing and one or more relatively simple heuristic procedures for packing such as deepest-bottom-left-fill. Used metaheuristic strategies for routing are among others TS (Bortfeldt, 2012; Gendreau et al., 2006; Wisniewski, Ritt, and Buriol, 2011; Zhu, Qin, Lim, and Wang, 2012), a combination of TS and guided local search (Tarantilis, Zachariadis, and Kiranoudis, 2009), ant colony optimization (Fuellerer, Doerner, Hartl, and Iori, 2010), a combination of GA and TS (Miao, Ruan, Woghiren, and Ruo, 2012), adaptive variable neighborhood search (Wei, Zhang, and Lim, 2014), evolutionary local search (Zhang, Cai, Ye, Si, and Nguyen, 2017) and column generation technique-based heuristics (Mahvash, Awasthi, and Chauhan, 2017). Some more elaborated and effective packing procedures were employed by Bortfeldt (2012), Gendreau et al. (2006), Tao and Wang (2015) and Zhang et al. (2017).

A model and exact approach of the 3L-CVRP was provided by Junqueira, Oliveira, Carravilla, and Morabito (2013); this model has additional constraints compared to the Gendreau formulation and only small instances with nodes up to 15 and boxes up to 32 were solved. Hokama, Miyazawa, and Xavier (2016) proposed two Branch-and-Cut algorithms for 3L-CVRP variant with just LIFO constraint, but no fragility and stability constraints. They solved instances like E101-14 s (Gendreau et al., 2006) with 100 nodes and 198 boxes.

Other features are also considered such as time windows by, e.g., Moura (2019) and Vega-Mejia, Montoya-Torres, and Islam (2019), Moura and Oliveira (2009), pallet loading by Song, Jones, and Asgari (2019), Zachariadis, Tarantilis, and Kiranoudis (2012) and Zhang et al. (2017), pallet loading and axle weight constraints by Pollaris, Braekers, Caris, Janssens, and Limbourg (2017), backhauls by, e.g., Reil, Bortfeldt, and Mönch (2018) and Koch, Bortfeldt, and Wäscher (2018), heterogeneous fleet by Pace, Turky, Moser, and Aleti (2015), pick-up and delivery by, e.g., Bartók and Imre (2011) and Männel and Bortfeldt (2016).

2.3. Approaches to 3L-SDVRP

To the best of our knowledge, there are only four works that involve the possibility of splitting the customers' demands in a routing-packing problem context.

Moura and Oliveira (2009) proposed two methods for the vehicle routing problem with time windows and 3D loading constraints. In their sequential method, a sequential candidate list (SCL) is defined and a candidate of SCL is composed of a client and a single box type of his demand. When none of the remaining candidates in the SCL can feasibly be assigned to the current vehicle, a new, empty vehicle is opened; thus, a candidate with the same client and a different box type might be assigned to a different vehicle. This procedure can be viewed as the initial approach to split delivery in the 3L-CVRP context, however, the split delivery is only done temporarily as a step to final non-split solutions.

Ceschia et al. (2013) deal with three problems: first, the 3L-CVRP in Gendreau formulation; second, an extended 3L-CVRP with difficult packing constraints (load bearing strength, robust stability, and reachability); and third a 3L-SDVRP with same packing constraints as in the second one. Ceschia et al. tackle all three problems by a single-staged local search approach. Good results are achieved for well-known benchmark instances for the 3L-CVRP. To test their heuristic, they use 13 instances derived from practice and with limited vehicle fleet. The local search approach does not reach feasible solutions for all instances of the extended 3L-CVRP. In case of their 3L-SDVRP feasible solutions are provided, i.e. split delivery helped to achieve the missing feasible solutions. However, the results with split delivery are worse than those without split delivery, which is in contrast to the classical SDVRP. Furthermore, relatively long running times of up to 10,000 seconds are reported.

Yi and Bortfeldt (2018) address the 3L-SDVRP with the same packing constraints as the 3L-CVRP in Gendreau formulation. Only inevitable splits are allowed, i.e. serving a customer in two or more routes is only permitted if not all boxes can be packed into a single loading space. A hybrid heuristic is developed that can be considered as preliminary variant of the algorithm presented here.

Finally, Li, Yuan, Chen, Yao, and Zeng (2018) propose a novel data-driven three-layer search algorithm to solve the 3L-SDVRP. They minimize the number of used vehicles with first and the total travel distance with second priority. Also, Li et al. add a packing material constraint and other process constraints for their industrial scenario in which an average delivery order contains more than 300 boxes to be distributed across the Pearl River Delta region. Unfortunately, very small instances with only six customers are used for testing their approach.

To sum up the merits and shortcomings of the existing papers on 3L-SDVRP let us state that the paper by Moura and Oliveira (2009) remains in the domain of 3L-VRPTW (splitting deliveries is just an intermediate process). Ceschia et al. (2013) create a strong heuristic that is applied to problems with strongly and weakly heterogeneous box sets. But they are not able to prove any advantage of splitting deliveries. Yi and Bortfeldt (2018) only deal with forced splits (see Section 3.1) while Li et al. (2018) consider specific constraints but do not test their approach sufficiently. All in all, it has not been examined sufficiently whether splitting deliveries can be advantageous in the 3D case.

3. Essential features and problem variants and formulation of the 3L-SDVRP

We consider first essential features and problem variants of the 3L-SDVRP. A complete problem formulation is given below in Section 3.2.

3.1. Essential features and problem variants of the 3L-SDVRP

The 3L-SDVRP is basically defined as the 3L-CVRP. However, the customer in 3L-SDVRP can be visited more than one time by split delivery. There are two reasons to split the delivery of a customer: the splitting can be forced and it can be done optionally to gain a benefit. We consider both variants of splitting in detail.

3.1.1. Forced splitting

Forced splitting occurs if the load of a customer, given by a set of boxes, cannot be stowed in a single vehicle in terms of volume or weight. In the one-dimensional (1D) SDVRP splitting is forced if and only if the demand of a customer exceeds the volume or weight capacity of a vehicle. Thus, splitting only depends on the data of a given SDVRP instance. In the 3L-SDVRP the same applies with regard to the weight of a customer's demand. However, the volume of a customer's demand as reason for (forced) splitting is a little more difficult. Of course, if the total volume of a customer's demand is larger than the volume of the loading space of a vehicle, splitting has to be done (such customers are called *big nodes* afterwards). Let β_i be the ratio of the demand's volume d_i of customer i and the loading space volume Q (in%, $i = 1,\dots, n$). If β_i is less than 100%, the necessity of splitting depends not only on the instance data. For example, if β_i = 85% for customer i, a weaker packing algorithm might be unable to stow all boxes while a more elaborated packing algorithm can do and avoid splitting in this case. Hence, the necessity of splitting in the 3D case does depend on the instance data but also on the packing algorithm employed. Correspondingly, the set of big nodes of an instance is in general a *proper* subset of the set of customers where splitting is inevitable.

Table 1
Variants of (1D-) SDVRP following Irnich et al. (2014).

Variant	Description
SDVRP	Demands: $q_i \le Q$, $i = 1,\dots, n$. Visits: the number of visits to a customer can be any positive integer.
SDVRP$^+$	Demands: at least one customer j has a demand $q_j > Q$. Visits: the number of visits to a customer can be any positive integer.
VRP$^+$	Demands: at least one customer j has a demand $q_j > Q$. Visits: the number of visits to a customer i is given by $\lceil q_i / Q \rceil$ and the minimum delivery amount (MDA) per visit is given by (q_i mod Q).
H^{SDVRP+}	As SDVRP$^+$; in addition, it is assumed that each demand $q_i > Q$ is first split according to $q_i = \lfloor q_i/Q \rfloor \cdot Q + (q_i \bmod Q)$. Customer i is then served $\lfloor q_i / Q \rfloor$ times with a full load Q in a direct trip. The remaining problem with demands (q_i mod Q) has to be solved as SDVRP.
H^{VRP+}	As H^{SDVRP+} but for the remaining demands the MDA is given as in VRP$^+$, i.e. the remaining problem has to be solved as CVRP.

In the problem variant 3L-SDVRP-f ("f" as *forced*), splitting customer deliveries is only allowed if the demand of a customer cannot be stowed in a single vehicle by means of the used packing algorithm. Of course, this definition is extended naturally to the case where the load of a customer does not fit in two vehicles etc., i.e. only the minimum number of demand splits per customer is permitted. If for each customer all boxes can be stowed in the same vehicle then the problem is solved as 3L-CVRP. The subtype 3L-SDVRP-f corresponds to a management point of view that emphasizes the additional organizational and working power effort caused by splitting deliveries. Hence, splits should be avoided whenever possible while the number of routes (v) and the total travel distance (*ttd*) are to be reduced by other means than by splitting demands.

3.1.2. Optional splitting

Optional splitting occurs if the load of a customer fits in a single vehicle (in terms of volume and weight), but splitting is done nevertheless because it is beneficial to serve the customer in two or more routes, i.e. v and *ttd* can be reduced by splitting.

In the problem variant 3L-SDVRP-o ("o" as *optional*), splitting customer deliveries can be done any number of times. Again, the judgment whether a solution for a 3L-SDVRP instance is a 3L-SDVRP-o solution cannot be done without considering the used packing algorithm. The subtype 3L-SDVRP-o corresponds to a management point of view whereupon splitting might be used as a means of reducing the total costs given by the *sum* of splitting costs and routing costs. Of course, optional splitting is worthless if it leads to increasing routing costs.

Irnich et al. (2014) present several variants of the (1D-)SDVRP that are used for worst-case analyses. Some of them are quoted in Table 1 as we would like to find relations to useful variants of the 3L-SDVRP.

In the SDVRP the vehicle fleet is mostly assumed to be homogeneous and unlimited, and we will *only* consider such variants. The number of customers is denoted by n, the vehicle capacity by Q and customer demands by q_i, $i = 1,\dots, n$. Based on 1D-SDVRP variants indicated in Table 1 and the above definitions, four variants of the 3L-SDVRP are listed in Table 2.

As indicated in Table 2 the 1D variants SDVRP and SDVRP$^+$ are integrated in the 3D case because of the "fuzzy" character of customers' demands that are to be split. The variants H$^{3L\text{-}SDVRP\text{-}o}$

Table 2
Variants of 3L-SDVRP.

Variant	Description	Related 1D-SDVRP variant
3L-SDVRP-o	Optional splitting is allowed besides forced splitting.	SDVRP, SDVRP$^+$
3L-SDVRP-f	Only forced splitting is allowed and is necessary for at least one customer.	VRP$^+$
$H^{3L\text{-}SDVRP\text{-}o}$	For each customer direct trips are to be planned (where necessary) until the residual demand can be packed in one vehicle. The residual problem has to be solved as 3L-SDVRP-o.	H^{SDVRP+}
$H^{3L\text{-}SDVRP\text{-}f}$	As $H^{3L\text{-}SDVRP\text{-}o}$, but the residual problem has to be solved as 3L-CVRP.	H^{VRP+}

and $H^{3L\text{-}SDVRP\text{-}f}$ represent heuristic approaches and specify that demands to be split are partially to be delivered by direct trips.

3.2. Problem formulation

In the 3L-SDVRP, n customers and a single depot (with index 0) are given. Each customer has a set I_i of boxes with known dimensions that are to be transported from the depot to the customer ($i = 1, \ldots, n$). The set I_i includes m_i boxes I_{ik} ($k = 1, \ldots, m_i$) and box I_{ik} has length l_{ik}, width w_{ik}, and height h_{ik} ($i = 1, \ldots, n$, $k = 1, \ldots, m_i$).

Let $V = \{0, 1, \ldots n\}$ denote the set of all customers (also called nodes) including the depot. Let E be a set of undirected edges (i, j) connecting all node pairs ($0 \le i < j \le n$) and let $G = (V, E)$ be the resulting graph. Let a symmetric distance d_{ij} ($d_{ij} \ge 0$) be assigned to each edge (i, j) ($0 \le i < j \le n$). Finally, there is an unlimited fleet of identical vehicles with a rectangular loading space with length L, width W, and height H. Each vehicle is rear-loaded.

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-axis, respectively. 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 box corner that is closest to the origin of the coordinate system; in addition, an orientation index o_{ik} indicates which of the possible spatial orientations is selected ($i = 1, \ldots, n$, $k = 1, \ldots, m_i$). 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. Fig. 1 shows a loading space with placed boxes.

A route R is a sequence $(0, c_1, \ldots, c_p, 0)$ that starts and ends at the depot. It is feasible if it includes $p \ge 1$ pairwise different customers ($0 < c_i \le n$, $i = 1, \ldots, n$). A solution of the 3L-SDVRP is a set of v ($v \ge 1$) pairs (R_μ, P_μ), where R_μ is a route and P_μ is the related packing plan ($\mu = 1, \ldots, v$). It is called feasible if the following conditions hold:

(F1) All routes R_μ and all packing plans P_μ are feasible ($\mu = 1, \ldots, v$).

(F2) Each customer occurs at least once in the routes R_μ ($\mu = 1, \ldots, v$); in problem variant 3L-SDVRP-f the number of occurrences of a customer i is limited by the minimum number of vehicles needed to pack the set of boxes I_i ($i = 1, \ldots, n$); in variant 3L-SDVRP-o the number of occurrences of a customer i is not limited ($i = 1, \ldots, n$).

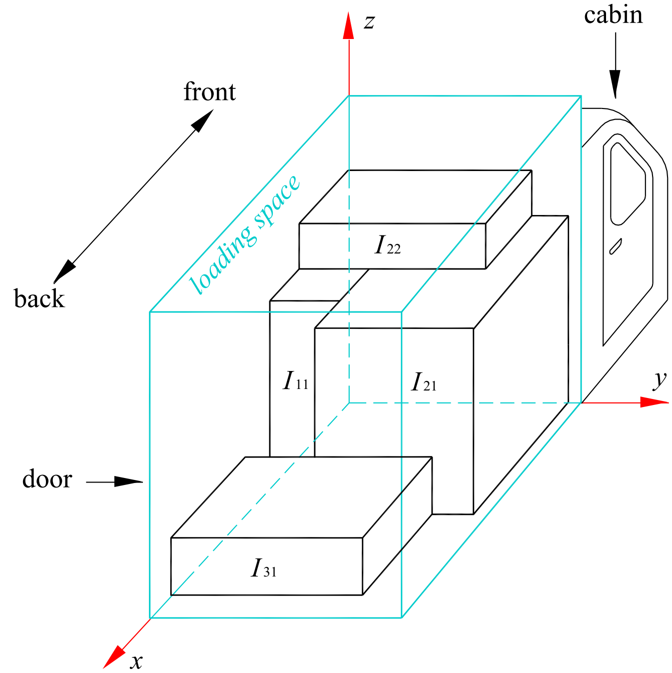

Fig. 1. A loading space with placed boxes.

(F3) The packing plan P_μ contains only placements of boxes that belong to customers visited in route R_μ ($\mu = 1, \ldots, v$); each box I_{ik} ($i = 1, \ldots, n$, $k = 1, \ldots, m_i$) is packed in exactly one packing plan P_μ ($\mu = 1, \ldots, v$).

In addition, the weight constraint and following loading constraints are integrated:

(C1) Weight Limit: Each box I_{ik} has a positive weight db_{ik} ($i = 1, \ldots, n$, $k = 1, \ldots, m_i$) and the total weight of all boxes in a packing plan P_μ must not exceed a maximum load weight D ($\mu = 1, \ldots, v$).

(C2) LIFO Policy: Let c and c' be two customers and c is visited before c' in route μ ($\mu \in \{1, \ldots, v\}$); let b and b' two boxes that belong to c and c', respectively. Then b' cannot be placed in packing plan P_μ between b and the rear of the vehicle or above b.

By this constraint it is ensured that all boxes of each customer can be unloaded by pure shifts in x-direction without moving other boxes.

(C3) Fixed Vertical Orientation: 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 ($i = 1, \ldots, n$, $k = 1, \ldots, m_i$).

(C4) Vertical Stability: If a box is not placed on the vehicle floor, a certain percentage a of its base area has to be supported by other boxes.

(C5) Fragility: 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 ($f_{ik} = 0$).

Finally, the 3L-SDVRP consists of determining a feasible solution that meets the constraints (C1) – (C5) and minimizes first the number of routes and second (with lower priority) the total travel distance. The formulation applies to all variants of the 3L-SDVRP listed in Table 2, where obvious conditions are to be added for the variants $H^{3L\text{-}SDVRP\text{-}o}$ and $H^{3L\text{-}SDVRP\text{-}f}$.

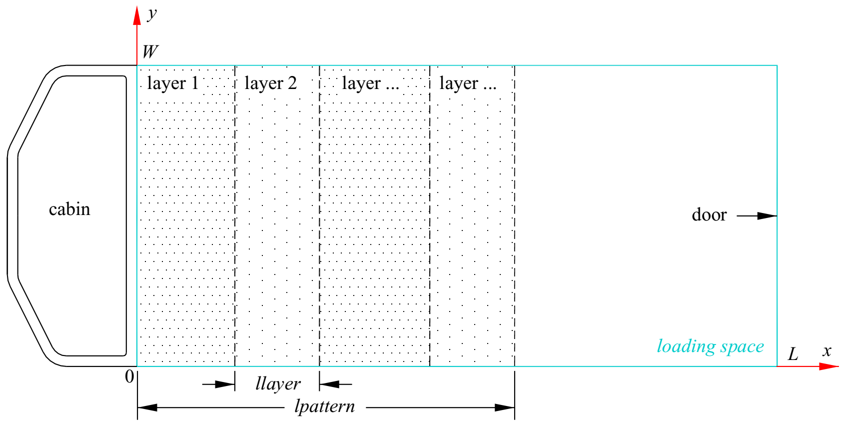

Fig. 2. Loading space and packing pattern (top view).

4. Hybrid algorithm

In this section, the hybrid algorithm for solving the 3L-SDVRP is described. It can be applied to both problem variants, namely the variant with forced splits and with optional splits.

4.1. Basic approach

The hybrid algorithm works in two main steps, the packing step and the routing step, which are done strictly one after another, not in an interlocked fashion.

4.1.1. The packing step

In the packing step, 3D packing patterns are generated taking into account the entire demand of all customers. Each 3D packing pattern fits in a single loading space and consists of one or more vertical layers or walls that follow one another along the length of a loading space. The width and height of a layer are given by the width and height of the loading space, respectively. The layer length *llayer* is given by the difference of the largest and the smallest x-coordinate of the layer. Both coordinates have to be taken by at least one box of the layer. The pattern length *lpattern* is given as the sum of the lengths of all layers in the pattern. A packing pattern with some layers is depicted in Fig. 2. For each customer one segment pattern (1C-SP) *must* and some further full load patterns (1C-FLPs) *can* be built. If multiple packing patterns are built the pattern with the lowest filling rate is taken as 1C-SP and the others are taken as 1C-FLPs. Together the patterns of a customer include all demanded boxes.

Big nodes have at least one 1C-FLP and also other customers can have one or more 1C-FLPs, which are then used for direct trips (see below). A segment pattern fills a loading space in general only partly. The segment patterns of multiple customers can be combined in one loading space if and only if the sum of the pattern lengths of all related segment patterns does not exceed the loading space length L.

Often the filling rates of segment patterns are relatively low since only boxes of a single customer are available for packing. Therefore, additional segment patterns, which store all boxes of two chosen customers, are built and termed 2C-SP patterns. Again, a 2C-SP consists of vertical layers, but now there is a layer where boxes of two customers can be placed. Later the two individual segment patterns of two customers can be replaced by the related 2C-SP. This can be done if the customers are visited one after another and might be advantageous if the length of the 2C-SP is smaller than the sum of the pattern lengths of the related 1C-SPs. In Fig. 3 the pattern types 1C-SP and 2C-SP are illustrated.

The boxes in the upper 1C-SP all belong to customer 1 while the boxes in the lower 2C-SP belong to customers 3 and 2 and there is a mixed layer with boxes from both customers. Since the customers of a 2C-SP have to be visited in direct succession, only pairs of customers are considered which are not too far apart from each other. For a customer c_1 only the *qnb*% of all customers c_2 are considered for 2C-SP (and called neighbors of c_1) which lead to the smallest insertion costs given as $d_{01}+d_{12}-d_{02}$ (0: depot, *qnb*: a parameter). A 2C-SP for the pair (c_1, c_2) is constructed so that c_1 can be visited first and c_2 can be visited second regarding constraint (C2).

All generated patterns (1C-FLP, 1C-SP and 2C-SP) are made available to the routing step. In the generation of packing patterns, the geometrical constraints (FP1)–(FP3) and the packing constraints (C3)–(C5) are observed. The routing step is mainly responsible for the weight constraint (C1) and the LIFO policy (C2). The generation of patterns is carried out by a genetic algorithm (1C-FLP, 1C-SP) and two construction heuristics (2C-SP).

4.1.2. The routing step

In the routing step first direct trips (depot→customer→depot) are constructed if there are full load patterns for one or more customers. Afterwards a reduced problem remains to be solved that can be characterized as follows:

- All boxes stowed in 1C-FLP and delivered by direct trips must not be considered any longer.
- Having packed all demanded boxes in packing patterns with a given pattern length only a one-dimensional VRP remains to be solved since the packing patterns (1C-SP, 2C-SP) can now be replaced by their pattern lengths.
- Hence, in this one-dimensional VRP two capacity restrictions are to be observed: the weight constraint (C1) and the length capacity constraint; the latter results from the packing step and requires that the sum of the pattern lengths of all 1C-SPs and 2C-SPs transported in the same route must not exceed the length L of the vehicle's loading space.

So far only *forced splits* have been implemented where appropriate and are represented for a given customer by his/her 1C-FLPs. Each of the 1C-SP and 2C-SP patterns can be accommodated in a single loading space. Thus, no further splits are necessary. Therefore, in the problem variant 3L-SDVRP-f only a one-dimensional CVRP remains to be solved in the routing step. This is done by a local search procedure. In problem variant 3L-SDVRP-o optional splits can be carried out and thus a one-dimensional SDVRP remains to be solved.

Optional splits are done as follows. Often the segment patterns of the customers of a route do not completely exhaust length L.

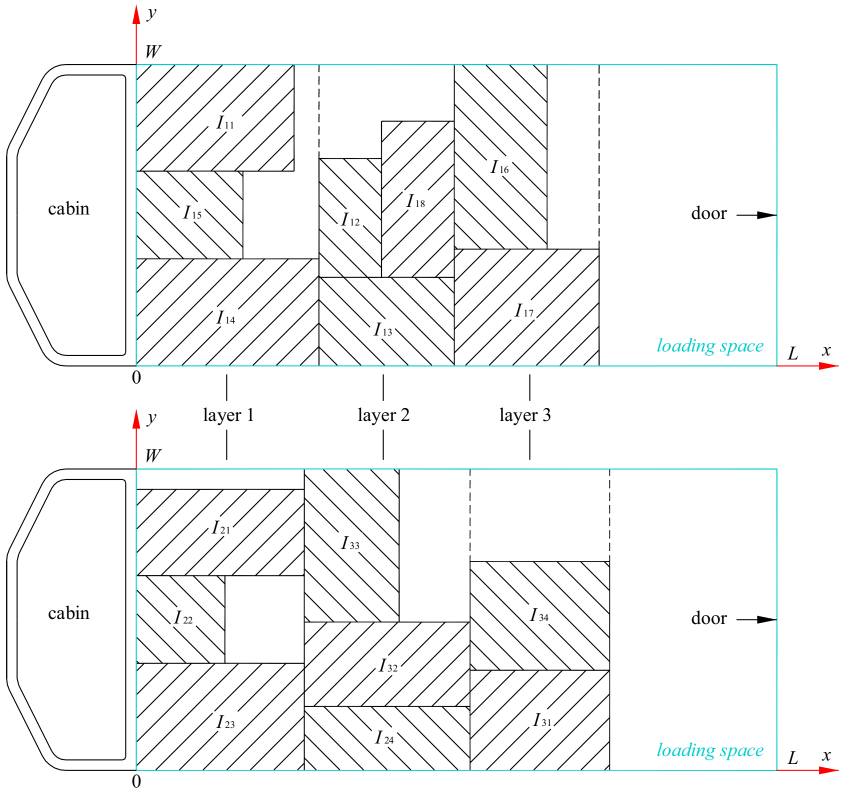

Fig. 3. Pattern types 1C-SP and 2C-SP.

There is a residual free length that, however, cannot be filled by all layers of another customer *c* not being a member of the route yet. In such cases it can be tried at least to pack a (proper) subset of the layers and to include customer *c* in this route. Of course, customer *c* must also be a member of another route whose loading space must accommodate the remaining layers of *c*. Thus, a "split customer" is always the last customer in a route and the first customer in the next route. Sometimes one or more of such splits allow to save an entire route and/or enable some distance saving. To carry out optional splits the local search procedure is modified appropriately utilising the layer structure of packing patterns.

4.1.3. Overview of the approach

The hybrid algorithm is summarized in Algorithm 1. After packing and routing the 3L-SDVRP solution is completed in the final step (not mentioned before), i.e. direct trips and all other routes of the best achieved routing solution are combined with related 3D packing patterns.

We would like to characterize the approach as follows. The packing and the routing task of the 3L-SDVRP are solved in separate steps following the principle “Packing first, Routing second” that has been successfully used recently by Reil et al. (2018) and ensures small running times. The structure of the generated packing patterns is very simple. Boxes are placed in vertical layers and each layer accommodates only boxes of one or two customers. Finally, for both 3L-SDVRP variants the algorithm follows the heuristic approach where customers with large demands are served by direct trips before the residual problem is tackled, i.e. the hybrid algorithm belongs to the type $H^{3L\text{-}SDVRP\text{-}f}$ or $H^{3L\text{-}SDVRP\text{-}o}$ (see Table 2). In the following, both main steps of the hybrid algorithm are considered in greater detail.

Algorithm 1 Overview of the hybrid algorithm.

```
SDVRLH2 (in: problem data, parameters, out: best solution s_best)
    // Packing step
    for each customer i do
        generate patterns for customer i (1C-FLPs where necessary, one 1C-SP)
        by container loading GA
    endfor
    for selected customer pairs (i,j) do
        generate 2C-SP pattern for customer pair (i, j) by construction
        heuristics
    endfor
    // Routing step
    for each customer i
        generate as many direct trips 0→i→0 as 1C-FLPs for customer i do
        exist
    endfor
    if only forced splits allowed then
        solve remaining CVRP by local search
    else // optional splits allowed
        solve remaining SDVRP by (modified) local search
    endif
    // Final step
    prepare solution s_best consisting of best achieved routing plan and
        related 3D packing patterns
end.
```

4.2. Generating packing patterns

The generation of packing patterns is done in three steps. In a preprocessing step stacks of boxes are built from the boxes of all customers. Patterns for all customers are then created by placing directly stacks instead of original boxes. Patterns of types 1C-FLP and 1C-SP are built in the second step while 2C-SP patterns are constructed in the third step.

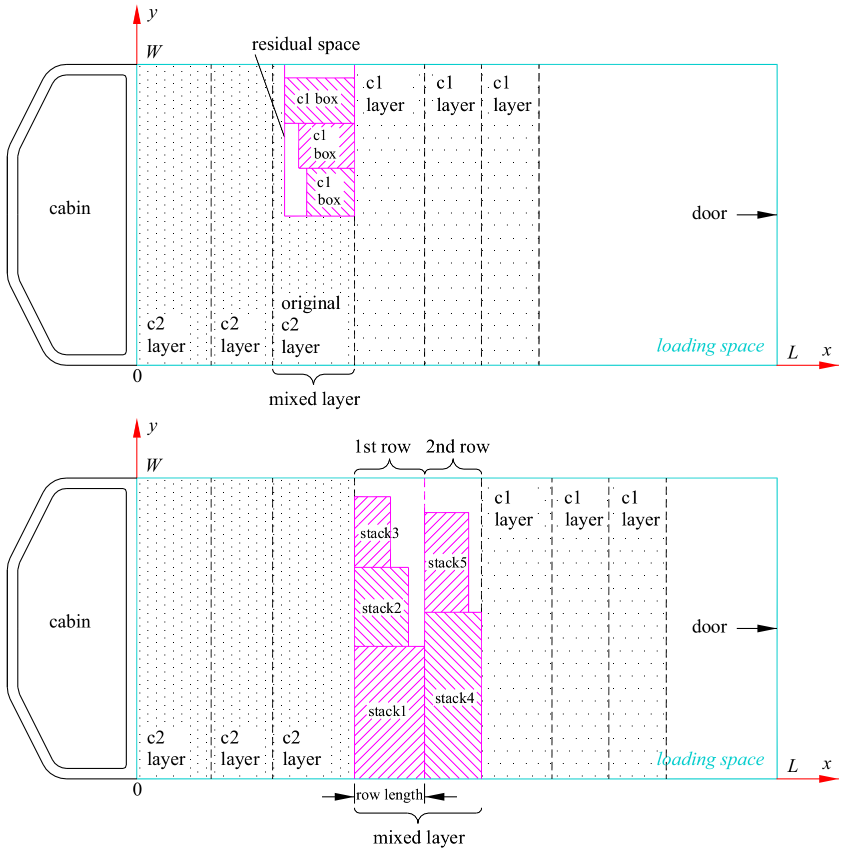

Fig. 4. Two 2C-SPs generated by two different procedures.

4.2.1. Building patterns of types 1C-FLP and 1C-SP

For each customer 1C-FLP and 1C-SP patterns are generated separately. The patterns for an individual customer are produced one by one. Each time another vehicle loading space is filled using the GA for container loading by Bortfeldt and Gehring (2001). All boxes of the customer that are not yet packed are handed over to the GA at the next call. At last the patterns are marked as 1C-SP or 1C-FLP patterns according to the above-mentioned rule.

The GA generates patterns that consist of vertical layers following one after another along the length of the loading space (see Fig. 1). Patterns (complete solutions) are generated by means of problem specific genetic operators, namely a crossover and two variants of mutation. All operators firstly copy layers with high filling rates from parents to descendants. Since the latter are mostly still incomplete afterwards, new layers are added then by a construction heuristic. A single layer is generated in two steps. First the layer (i.e. its depth) is defined before it is filled by one or more boxes. Each box is placed in a corner of a cuboidal residual space and then three new residual spaces inside the one just filled are constructed and filled later if possible. All relevant constraints of the 3L-SDVRP are met. Moreover, up to five residual spaces per layer that could not be filled are provided here as a solution component to fill them later by boxes of another customer.

4.2.2. Building patterns of type 2C-SP

For a customer c_1 and a neighboring customer c_2 two different procedures are applied to construct 2C-SPs that are based on the 1C-SPs for both customers. Seen from the rear, first some layers of customer c_1 (c_1-layers for short) are chosen from the 1C-SP of c_1; then a (newly constructed) mixed layer with boxes from c_1 as from c_2 follows and the end of the pattern (nearer to the cabin) is formed by some layers of the 1C-SP for c_2. Fig. 4 shows a 2C-SP generated by the first (second) procedure in the upper (lower) part.

The *first* procedure is based on the provided residual spaces of 1C-SPs. For each pair (c_1, c_2) a c_2-layer is selected and its residual spaces are taken to accommodate the boxes of a c_1-layer to form a mixed layer. If this works, the c_1-layer is saved and the procedure terminates.

In the *second* procedure, first c_1- and c_2-layers with the worst filling rates of the related 1C-SPs are removed. Then the boxes of the removed layers are taken to build stacks which are placed in one or more rows in a new mixed layer between the remaining c_1- and c_2-layers.

Both procedures ensure that LIFO constraint (C2) is met. At the end, for a customer pair (c_1, c_2) only the 2C-SP with the largest saving δ is accepted (if $\delta > 0$). The saving δ is defined as:

$$\begin{aligned}\delta(2C-SP(c_1,c_2)) := \; & lpattern(1C-SP(c_1)) \\ & + lpattern(1C-SP(c_2)) \\ & - \; lpattern(2C-SP(c_1,c_2)).\end{aligned} \quad (1)$$

4.3. Routing algorithm

The routing algorithm is based on a representation of solutions as giant tours (cf. Vidal, Crainic, Gendreau, and Prins, 2014, p. 667) where any solution is given as a permutation of all n customers without explicit mention of visits to the depot. Subsequently, we describe the decoding procedure and the main algorithm of the local search that is based on swap and shift moves. The problem variants 3L-SDVRP-f and 3L-SDVRP-o are handled in a by and large uniform manner.

4.3.1. Decoding a solution

The decoding procedure gets as input mainly a coded solution, i.e. a customer sequence, and returns the complete solution with (1) the routes of the solution, (2) all implemented customer pairs (CP), i.e. all pairs of consecutive customers for which a 2C-SP pattern is taken instead of two 1C-SP patterns, and (3) the customers whose delivery is split and the related layer distribution (only for problem variant 3L-SDVRP-o).

As mentioned above, a customer can be visited at most two times. If a customer is visited twice then he/she must be the last customer in a first route and the first customer in a second route. Besides the set of customers being visited twice a solution must include for each such customer the distribution of his/her cargo layers between both visits.

In the first step of decoding the CPs to be implemented are selected. For this purpose, the potential CPs, i.e. all pairs of customers (c_1, c_2) in the coded solution for which a 2C-SP pattern exists, are examined. Each customer can occur in multiple potential CPs while at most one pair per customer can be chosen for implementation. To maximize the total savings in terms of loading length a greedy approach is applied.

In the second step of decoding the routes are determined and decisions on splitting deliveries are made where necessary. For that purpose, the complete customer sequence is traversed. At each position it is first checked whether the related customer c is a single customer or the first customer of a CP (c, c') selected before.

If the customer c at the current position is a single customer, it is verified whether the associated 1C-SP fits completely into the vehicle of the current route in terms of weight and loading length. If yes, the current route is updated (weight, loading length) and assigned to c. Otherwise, a splitting check is done. The delivery of customer c is split to the current and next route if two conditions are satisfied. First, at least one layer of the 1C-SP must fit into the vehicle. Second, splitting the delivery must not be "too expensive" with regard to *ttd*. That is the following criterion regarding the relative insertion costs for customer c must be observed

$$(d_{cprev,c} + d_{c,0})/d_{cprev,0} \leq max\_split\_costs \tag{2}$$

where c_{prev} is the predecessor of c in the current route and max*_split_costs* is a parameter.

Note that by means of criterion (2) and value max*_split_costs* = 0 problem variant 3L-SDVRP-f is implemented, too. If the outcome of the splitting test is positive, the layers of customer c are examined in descending order of their loading lengths. Each layer that still fits into the current route is accepted for it while the other layers are reserved for the next route in which c is the first customer. If the result of the splitting test is negative, customer c (with the related 1C-SP) only occurs in the next route.

If customer c is the first one of a CP (c, c'), the decoding procedure runs similarly as for a single customer. Note that both customers of the CP are processed at once.

4.3.2. Main algorithm of local search

First an initial solution for the local search is determined by a randomized variant of the Savings heuristic. The local search is then organized in multiple partial searches (see Algorithm 2). Each partial search is characterized by two features. First, by the maximum admissible splitting costs that are taken into account. Second, by the range of swap and shift moves given by parameter *nbh_size*. Thus, small and large ranges alternate to diversify the search again.

The possible values of the admissible maximum splitting costs are held in the vector *Max_split_costs*. At the first place the value 0 is recorded, saying that no optional splitting is allowed at all. Hence, the parameter *nps* is set to 1 for problem variant 3L-SDVRP-f and in this case only two partial searches are performed. Only for problem variant 3L-SDVRP-o further values of *Max_split_costs* on places 2,3, … are used. The values of *Max_split_costs* show an exponential growth so that large splitting costs are accepted at the end.

Algorithm 2 Main algorithm of local search.

Algorithm local_search(in: problem data, parameters, out: best solution s_{best})
generate initial solution s_{init} and set s_{best}:= s_{init}
for ips:= 1 to nps **do** // outer loop: changing max. admissible splitting costs
for inbh:= 1 to 2 **do** // inner loop: changing range of moves
// specify max. admissible splitting costs
max_split_costs:= Max_split_costs[ips]
generate 2C-SP pattern for customer pair (i, j) by construction heuristics
// specify range size
if inbh = 1 **then** nbh_size = MAX(n/4, 3) **else** nbh_size = MAX(n, 3) **endif**
// partial search
s_{curr}:= s_{best} // starting from best solution so far
for iter = 1 to niter **do**
s_{iter_best}:= determine best neighbour of s_{curr} by swap/shift moves with range nbh_size
post-optimize s_{iter_best} by 2-opt
if ips = 1 and $n \leq 100$ **then** post-optimize siter_best by 3-opt **endif**
update best solution s_{best} by s_{iter_best} where necessary
s_{curr}:= s_{iter_best}
endfor
endfor
endfor
post-optimize s_{best} by 3-opt
end.

Each partial search is started from the best solution s_{best} achieved so far. Per iteration first the best neighbor of the current solution s_{curr} is calculated before the best neighbor solution is post-optimized by 2-opt and possibly by 3-opt (see, e.g., Helsgaun, 2009). Note that the decoding procedure has also to be integrated in the local search methods 2-opt and 3-opt. At the end of an iteration the best solution s_{best} is updated if necessary and s_{curr} is set to s_{iter_best} for the next iteration. Finally, 3-opt is applied again, this time to the best solution s_{best}.

5. Numerical experiments

In this section, we first describe the setting of the parameters used in our approach. We test three sets of instances, namely the Shanghai instances, the Ceschia instances and the new B-Y instances. The first two sets are derived from real-world operations while the B-Y instances are based on well-known VRP and CLP benchmark instances. All instances have weakly heterogeneous box sets that are common in the context of factory inbound logistics. We will describe the instance sets and compare our results with the best ones in the literature on the 3L-SDVRP. In addition, we will test the algorithm with the 27 3L-CVRP instances by Gendreau et al. (2006) which have strongly heterogeneous box sets. Our instances and results can be found in the supplementary material of this paper.

All experiments were run on a 3.40 GHz PC (AMD A10-5700 APU) with 8.0 GB RAM under Windows 8.1. Our algorithm, denoted by SDVRLH2, was coded using MS Visual C++ 2010 (Express Edition). Each instance of the first three sets was run ten times while the instances by Gendreau et al. (2006) were run five times.

5.1. Parameter setting

The parameters for the GA that serves the generation of 1C-SP and 1C-FLP patterns are chosen as in Bortfeldt and Gehring (2001).

Table 3
Parameters for packing and routing.

Parameter	Meaning	Value
qnb	Percentage of neighboring customers (in%)	30
maxcands	Maximum no. of variants for next stack placement	if $n \leq 50$: 3 else: 2
nindiv	No. of trials for generating the initial solution	10
nbh_size (small)	Small range size for swap / shift moves	MAX(n/4, 3)
nbh_size (large)	Large range size for swap / shift moves	MAX(n, 3)
niter	No. of iterations in a partial search	100
niter_wimpr	No. of iterations without improvement; a partial search is terminated if *niter_wimpr* iterations are carried out without yielding a new best solution s_{best}	20
maxtime	Time limit for entire search (in s)	if $n < 50$: 240 else if $n < 200$: 1800 else: 3600
maxtime-2opt	Time limit for single 2-opt run (in s)	120

Further parameters for packing and routing are set as depicted in Table 3 (n stands for the number of customers).

The number of partial searches *nps* and the vector *Max_split_costs* are determined as follows. First the minimum and maximum relative insertions costs ric_{min} and ric_{max} over all pairs of customers are calculated according to $ric_{min} = \text{MIN}\ \{(d_{ij} + d_{j0})\ /\ \text{d}_{i0}\ \mid 1 \leq i, j \leq n\}$ and $ric_{max} = \text{MAX}\ \{(d_{ij} + d_{j0})\ /\ \text{d}_{i0}\ \mid 1 \leq i,j \leq n\}$. Then we set *Max_split_costs*(1) = 0 and *Max_split_costs*(2) = $2 \cdot ric_{min}$. On each of the following places 3,4,…the preceding value is doubled until the value ric_{max} is exceeded. Then the last value is replaced by ric_{max}. The number of occupied places of *Max_split_costs* gives the number of partial searches *nps*. The specified parameter setting was determined in pre-tests of limited size and is used for all numerical experiments.

5.2. Experiments with Shanghai instances

Our data come from the cargo collecting operations in and around Shanghai area by a Shanghai automotive logistics company, which serves many car makers in Shanghai and all over China. Both the data of routing aspects or packing aspects are from real-world.

There are 3849 order lines in the data table, each with fields like part number, pick-up node address, package dimensions ($h \times w \times l$), and required number etc. In summary, there are 200 pickup nodes that are located around Shanghai area with each node-pair distance within 120 km. These parts in all have 634 package dimensions ($h \times w \times l$) with the smallest box $6 \times 9 \times 12$ Cubic Centimetre and the largest box $190 \times 155 \times 225$ Cubic Centimetre. For each pickup node, there are 24 items on average to pickup with the highest number 270 boxes.

For our Shanghai dataset, based on the areas, we divided the whole data table into 11 first-level instances with node addresses ranging from 5–46 (Yi and Bortfeldt, 2018). The box number of each instance ranges from 73–1459. In the second level, these 11 instances are combined to 3 big ones by area amalgamation. Finally, in the third level, Sha15 combines all eleven instances (Sha1–11) together with total 200 nodes, 634 box types and 4776 boxes. These 15 instances are summarized in Table 4.

A unique character of the Shanghai instances is that there are *big nodes* with $\beta > 100\%$. In Fig. 5 the distribution of β values of the 200 nodes in Sha15 is depicted. In practice, these big nodes are usually secondary collecting points which receive parts or components from suppliers far from Shanghai area. To pick up demand from big nodes, split delivery has to be applied. However, whether to split other nodes leads our problem to both 3L-SDVRP-f and 3L-SDVRP-o categories.

There are four vehicle types for different instances in Table 4: SV with $H \times W \times L$ dimensions $175 \times 180 \times 415$ Cubic Centimetre, weight capacity 2.5t; MV: $242 \times 217 \times 640$ Cubic Centimetre, 8 t; BV: $240 \times 240 \times 860$ Cubic Centimetre, 15 t; CV: $270 \times 235 \times 1202$ Cubic Centimetre, 26.46 t, respectively. However, for each instance, there is only one vehicle type, i.e. an unlimited homogeneous fleet.

The cost to apply a new vehicle is quite high, thus the objective priority is given to v, then to *ttd*. The lower bound *LB* on vehicles is calculated according to

$$LB = \left\lceil \sum_{i=1}^{n} \beta_i \right\rceil = \left\lceil \sum\nolimits_i d_i / Q \right\rceil = \left\lceil \sum_{i=1}^{n} \sum_{k=1}^{m_i} h_{ik} w_{ik} l_{ik} / HWL \right\rceil \tag{3}$$

In each instance, the pickup node addresses and their distances are gained from a public e-map. For the Shanghai instances the Gendreau formulation of the 3L-SDVRP is assumed, i.e. the constraints (C1) – (C5) have to be observed. The percentage a of stability constraint (C4) is set to 75% (and the same is done for the other instance sets).

The results for the Shanghai instances are presented in Table 5. For the problem variants with forced and optional splitting average values of v and *ttd* and average run times *rt* (in seconds) as well as best values of v and *ttd* over ten runs are indicated. In the last two lines sums are given for v and averages for *ttd* and *rt*.

Table 4
Summary of the Shanghai instances.

Instance	Nodes (n)	Box types (bt)	Items (M)	Vehicle type	Lower bound LB on vehicles*	Items per box type	Items per customers
Sha1	5	26	261	MV	2	10.0	52.2
Sha2	8	50	167	SV	6	3.3	20.9
Sha3	10	17	73	SV	3	4.3	7.3
Sha4	12	33	204	BV	3	6.2	17.0
Sha5	12	59	228	MV	4	3.9	19.0
Sha6	15	56	228	MV	4	4.1	15.2
Sha7	16	79	439	BV	7	5.6	27.4
Sha8	18	51	303	MV	6	5.9	16.8
Sha9	27	98	734	CV	8	7.5	27.2
Sha10	31	134	590	BV	9	4.4	19.0
Sha11	46	185	1549	CV	16	8.4	33.7
Sha12	35	149	623	MV	10	4.2	17.8
Sha13	64	234	1494	BV	19	6.4	23.3
Sha14	101	311	2659	CV	26	8.5	26.3
Sha15	200	634	4776	CV	42	7.5	23.9

* : for calculation see Eq. (3) below.

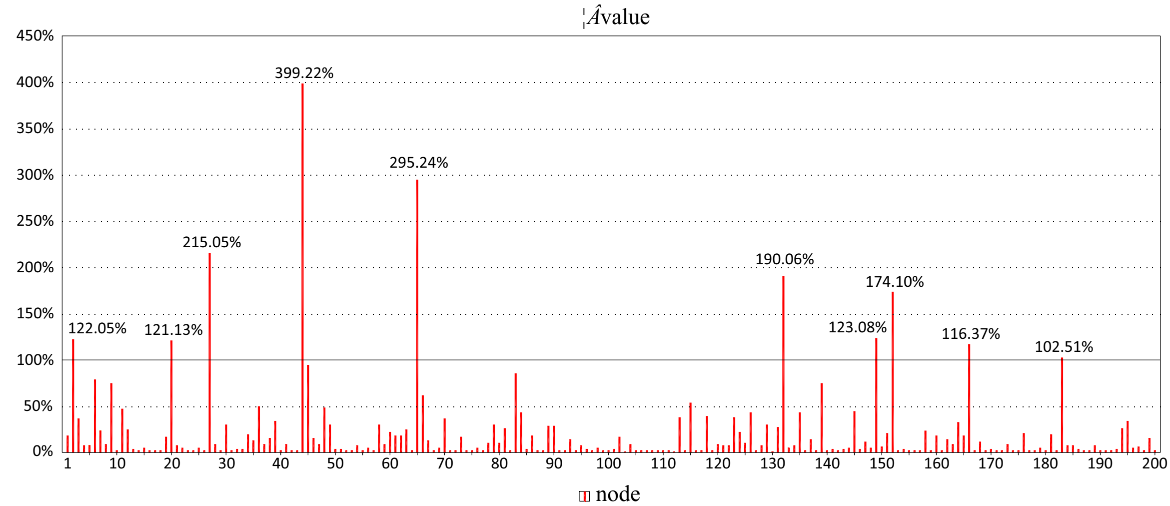

Fig. 5. Distribution of β value (in%) of the 200 nodes in instance Sha15.

Table 5
Results for Shanghai instances.

Instance	SDVRLH2, forced splitting					SDVRLH2, optional splitting				
	Avg		*Best*			*Avg*		*Best*		
	v	*ttd*	*rt (s)*	*v*	*ttd*	*v*	*ttd*	*rt (s)*	*v*	*ttd*
Sha1	3.0	562.6	0.1	3	562.6	3.0	562.6	0.3	3	562.6
Sha2	9.0	2813.2	11.8	9	2813.2	9.0	2786.6	11.5	**9**	**2786.6**
Sha3	4.0	397.7	0.4	4	393.0	4.0	396.5	1.1	4	393.0
Sha4	4.0	352.2	2.7	4	349.7	4.0	350.3	3.0	4	349.7
Sha5	6.0	1451.3	7.5	6	1447.6	6.0	1451.3	7.8	6	1447.6
Sha6	8.0	486.1	10.6	8	481.3	8.0	482.1	11.2	**8**	**479.6**
Sha7	12.0	1708.3	16.0	12	1694.8	12.0	1707.5	16.8	12	1694.8
Sha8	8.5	376.9	5.1	8	360.9	8.0	357.7	5.7	**8**	**352.7**
Sha9	13.1	955.1	12.2	13	942.6	13.0	944.3	16.9	**13**	**928.7**
Sha10	14.0	1610.2	15.8	14	1590.1	14.0	1605.4	20.5	14	1590.1
Sha11	27.0	518.2	93.2	27	512.6	26.2	523.2	150.7	**26**	**521.5**
Sha12	17.0	2972.7	35.4	17	2942.1	17.0	2968.1	48.1	17	2942.1
Sha13	28.0	3174.6	424.8	28	3128.2	27.7	3147.9	594.1	**27**	**3097.3**
Sha14	45.1	1538.5	883.0	45	1481.8	44.2	1574.4	1802.3	**44**	**1522.5**
Sha15	72.1	4860.3	3620.5	71	4867.8	71.9	4867.0	3625.9	**71**	**4795.4**
Total	270.7	1585.2	342.6	269	1571.2	268.0	1581.7	421.1	**266**	**1564.3**

Compared to Yi and Bortfeldt (2018), our new algorithm SDVRLH2 improves the results in terms of *v* by 4.4% and *ttd* by 7.3% on average and needs on average 6 seconds less run time.

Looking at the results for the problem variants 3L-SDVRP-f and 3L-SDVRP-o, we can see that *v* was improved for 3L-SDVRP-o by 1.1% and *ttd* was reduced by 0.4% if best values are compared. With optional splits the run time is about 23% higher though. There are at least eight of 15 instances with better results (in **bold**) than those of 3L-SDVRP-f (note that the objective function is a vector). However, only a small improvement was achieved by optional splits.

The mean volume utilisation of the loading spaces over the 15 instances amounts to 57.4% in variant 3L-SDVRP-f and to 57.8% in variant 3L-SDVRP-o; this can be considered a good utilisation if the packing constraints are taken into account.

5.3. Experiments with instances from Ceschia et al. (2013)

Ceschia et al. (2013) present a set of industrial instances which exhibit a high variability in terms of the number of customers, the number of box types and the number of boxes. There are 13 instances with customers ranging from 11–129, boxes ranging from 254–8060, boxes per customer ranging from 1–217. These boxes have dimensions ranging from $10 \times 1 \times 11$–$239 \times 100 \times 230$ ($h \times w \times l$, Cubic Centimetre). For all instances the available number of vehicles is fixed (limited fleet), and some instances have a heterogeneous fleet. In their paper, the authors report on three experiments with their instances. Here we quote only their results for the first experiment regarding the 3L-CVRP in Gendreau formulation.

Since we want to minimize *v* and *ttd*, we have modified the Ceschia instances. We have kept the same nodes and boxes but have changed to an unlimited homogeneous fleet. For the original instances SD-CSS 5, 7, 8 and 11 with heterogeneous fleet we have chosen the smallest vehicle type with dimensions $268 \times 247 \times 1362$ Cubic Centimetre for our modified instances. We consider only the Gendreau formulation of 3L-CVRP and 3L-SDVRP.

The modified Ceschia instances are treated as 3L-SDVRP-f and 3L-SDVRP-o problems. The results are summarized in Table 6.

Direct trips as a result of forced splitting only occur for instance SD-CSS3, this is consistent with Table 2 of Ceschia et al. (2013). Moreover, for a comparison the results of their first experiment (3L-CVRP with Gendreau formulation) are also listed. Since splitting is not allowed in this case, there is no feasible solution of instance SD-CSS3 and this instance is *not* taken into account in the comparison (see line Total-12).

Compared to Ceschia et al. (2013) better results are reached for problem variant 3L-SDVRP-f. If the best 3L-SDVRP-f solutions are

Table 6
Results for (modified) Ceschia instances.

Instance	SDVRLH2, forced splitting					SDVRLH2, optional splitting					Ceschia et al., 2013		
	Avg		*Best*			*Avg*		*Best*			*Best*		
	v	*ttd*	*rt* (*s*)	*v*	*ttd*	*v*	*ttd*	*rt* (*s*)	*v*	*ttd*	*rt* (s)	*v*	*ttd*
SD-CSS1	5.0	5543.6	0.2	5	5467.4	5.0	5238.5	0.9	**5**	**4941.8**	351.2	*5*	*5084.1*
SD-CSS2	17.0	13,044.5	3.0	17	13,044.5	14.0	11,929.8	8.4	**14**	**11,620.2**	4709.1	*13*	*11,879.9*
SD-CSS3	24.0	16,327.6	21.1	24	16,154.0	23.0	16,302.1	36.4	**23**	**15,800.9**	–	–	–
SD-CSS4	14.1	12,958.5	13.3	14	12,700.3	14.0	12,826.7	27.0	**14**	**12,655.5**	5646.8	*12*	*11,175.2*
SD-CSS5*	17.0	14,513.8	19.5	17	14,513.8	17.0	14,513.8	33.5	17	14,513.8	10,000	*12*	*10,451.0*
SD-CSS6	17.9	15,850.9	36.7	*17*	*16,021.3*	17.0	15,883.7	70.2	**17**	**15,332.5**	1109.7	32	21,487.6
SD-CSS7*	13.2	12,330.1	47.3	13	12,084.8	13.0	12,206.6	73.0	**13**	**11,804.4**	6986.1	*10*	*10,339.7*
SD-CSS8*	25.0	19,747.2	44.3	*25*	*19,474.4*	23.0	19,709.8	123.3	**23**	**19,225.8**	4650.8	36	21,908.7
SD-CSS9	20.6	16,638.1	133.7	*20*	*16,585.7*	20.0	16,512.9	220.5	**20**	**16,015.0**	1694.6	23	17,258.0
SD-CSS10	18.4	16,380.4	205.1	18	16,079.4	18.0	16,432.6	273.8	**18**	**15,874.5**	9210.1	*18*	*11,865.1*
SD-CSS11*	21.1	18,871.3	1825.5	21	18,593.2	21.1	18,871.3	1822.7	21	18,593.2	1443.5	*13*	*24,843.1*
SD-CSS12	45.8	37,144.5	1135.0	*45*	*36,793.9*	44.0	37,261.3	1803.6	**44**	**36,311.5**	1734.7	48	34,256.1
SD-CSS13	25.0	21,977.5	1782.0	***25***	***21,338.6***	25.0	22,047.5	1802.2	25	21,694.3	5490.1	31	26,342.9
Total	264.1	17,025.2	405.1	261	16,834.7	254.1	16,902.8	484.3	**254**	**16,491.0**			
Total-12	240.1	17,083.4	437.1	237	16,891.4						4418.9	253	17,241.0

Note: Italic: better in comp. SDVRLH2_f vs. Ceschia et al. (2013); **bold**: better in comp. SDVRLH2_f vs. SDVRLH2_o.
* : instance with only one vehicle type, i.e. with the smallest type of the corresponding original instance.

compared (see Total-12), the mean improvement on *v* and *ttd* is 6.3% and 0.9% respectively. Moreover, the run times of SDVRLH2 are about one order of magnitude lower than those of Ceschia et al.

A comparison of our results for the problem variants 3L-SDVRP-f and 3L-SDVRP-o proves that for the (modified) Ceschia instances optional splits lead to better solutions. The improvement in terms of *v* and *ttd* is 2.7% and 3.1%, respectively, when the best solutions are compared. For 10 instances the best solution could be improved (marked bold in Table 6).

The comparison of the reached average solution quality does confirm this outcome. The average volume utilisations of the loading spaces in the best solutions amount to 44.1% (3L-SDVRP-f) and 45.6% (3L-SDVRP-o) and can again be seen as satisfactory (see 5.2). The increase of run times is about 19.6% with variant 3L-SDVRP-o.

5.4. Experiments with B-Y instances

To further test our algorithm, we have introduced a new set of 20 B-Y instances which are constructed by combining 5 CVRP instances from Christofides, Mingozzi, and Toth (1979) and 200 CLP instances by Bischoff and Ratcliff (1995). The first 5 CVRP instances C1-C5 with *n* ranging between 50 and 199 are chosen, and boxes are from CLP instances with 3 (test case BR1) and 20 box types (test case BR7). We use these two test cases in order to ensure two strongly different levels of box heterogeneity. For the loading spaces the dimensions of a 20 feet container are taken. The weight related data are derived from the demands and capacities of the CVRP instances. Each fifth box of a customer is set fragile.

For one-dimensional SDVRP instances the largest savings are obtained if the average customer demand is just above half of the vehicle capacity and the variance of the customer demands is low (Archetti and Speranza, 2012, p. 9). Therefore, we have constructed for ten instances customer box sets where the ratio β of the total box volume and the loading space volume is just above 50% for each individual customer. For the other ten instances we have generated box sets where β is circa 20% per customer, thus a tour has mostly less than five customers and there is a clear distinction to the first ten instances regarding β. A summary of the 20 B-Y instances is given in Table 7 (*M*, *bt* as in Table 4, β-mean represents the mean value of β over all customers).

The results for the B-Y instances are indicated in Table 8. Two columns include the mean filling rates *fr* of the vehicle loading spaces as percentages. In the last three lines sums are presented for the vehicle numbers *v* and averages for all other quantities. The lines Total-50 and Total-20 include the best *v*, *ttd* and *fr* values for the instance subgroups where β is circa 50% and 20%, respectively.

The main outcome here is that much better results are reached by optional splits. The improvements are 19% for the number of vehicles, 9.5%-points for the volume utilisation and 8.5% for the total travel distance if best solutions are compared. An improvement has been reached in 18 of 20 instances (marked bold in Table 8). Again, the comparison of average solutions confirms this result. The average run time for the variant with optional splits is about 47% higher than for the variant with forced splits. However, with circa 20 minutes on average the run times remain moderate for the variant with optional splits, too. As expected, forced splits proved to be not necessary for any instance.

For the subgroup with $\beta \approx 20\%$ the improvements are relatively small and the *ttd* values are even slightly worse if optional splits are admissible. However, in the other subgroup with $\beta \approx 50\%$ large improvements are made by optional splits (see line Total-50). For example, the average volume utilisation grows by 17.5%-points. Thus, the results for the 3D case are in line with the above quoted conclusion by Archetti and Speranza (2012).

While parameter β has a major impact on the results also the number of box types *bt* has a certain influence. There are three instances for which the number of vehicles is not improved by optional splitting (B-Y6, B-Y10 and B-Y18) and in all these cases we have $bt = 3$. This might be caused by the fact that for value $bt = 3$ there are much less valid packing patterns than for value $bt = 20$. In unfavourable situations regarding the box dimensions the positive effect of optional splitting can thus be paralyzed.

5.5. Experiments with instances by Gendreau et al. (2006)

The 27 3L-CVRP instances by Gendreau et al. (2006) correspond to the typical situation in outbound or distribution logistic with large numbers of relatively small orders and *strong* heterogeneous box sets where the number of box types equals the number of boxes. We undertake the test to push our algorithm that has been designed for *weekly* heterogeneous box sets to its limits. The results for variant 3L-SDVRP-f can be summed up as follows: the given maximum numbers of vehicles (or tours) are far exceeded. The total travel distances are on average by 55.9% larger than the distances reported by Gendreau et al. (2006). The average run time per instance amounts to 254 second. Small improvements are reached for variant 3L-SDVRP-o. On average 2.3% of the tours

Table 7
Summary of the B-Y instances.

Instance	*n*	*M*	*bt*	β-mean (%)	Instance	*n*	*M*	*bt*	β-mean (%)
B-Y1	50	3535	3	51.8	B-Y2	50	1295	3	19.0
B-Y3	50	3377	20	50.9	B-Y4	50	1442	20	22.0
B-Y5	75	5218	3	50.3	B-Y6	75	2135	3	21.6
B-Y7	75	5116	20	52.4	B-Y8	75	2054	20	20.5
B-Y9	100	6861	3	50.1	B-Y10	100	2854	3	21.4
B-Y11	100	6866	20	52.5	B-Y12	100	2680	20	20.7
B-Y13	150	10,356	3	51.2	B-Y14	150	4508	3	20.8
B-Y15	150	10,082	20	52.1	B-Y16	150	3904	20	20.7
B-Y17	199	14,230	3	51.5	B-Y18	199	5790	3	20.3
B-Y19	199	13,058	20	51.3	B-Y20	199	5797	20	22.2

Table 8
Results for B-Y instances.

Instance	SDVRLH2, forced splitting						SDVRLH2, optional splitting					
	Avg		*Best*				*Avg*		*Best*			*fr* (%)
	v	*ttd*	*rt* (*s*)	*v*	*ttd*	*fr* (%)	*v*	*ttd*	*rt* (*s*)	*v*	*ttd*	
B-Y1	50.0	2402.2	40.4	50	2402.2	51.8	37.0	2081.6	152.1	**37**	**2081.6**	70.1
B-Y2	16.0	1035.9	59.3	16	1019.4	59.3	15.5	1055.3	96.0	**15**	**1044.2**	63.2
B-Y3	50.0	2402.2	41.0	50	2402.2	50.9	37.0	2071.0	156.2	**37**	**2071.0**	68.7
B-Y4	19.1	1181.0	54.6	19	1163.9	57.9	18.0	1184.6	94.7	**18**	**1149.6**	61.1
B-Y5	75.0	3630.5	270.8	75	3630.5	50.3	54.0	2961.2	760.5	**54**	**2961.2**	69.8
B-Y6	25.9	1582.8	323.6	25	1573.5	64.8	25.0	1570.7	448.6	**25**	**1541.4**	64.8
B-Y7	75.0	3630.5	271.3	75	3630.5	52.4	57.0	3215.8	838.5	**57**	**3215.8**	69.0
B-Y8	27.1	1661.5	346.8	27	1643.6	56.9	26.0	1645.0	503.5	**26**	**1606.8**	59.0
B-Y9	100.0	4989.8	1064.1	100	4989.8	50.1	71.0	4014.9	1802.6	**71**	**4014.9**	70.5
B-Y10	34.7	2133.5	1817.3	34	2077.3	63.1	34.7	2133.5	1814.8	34	2077.3	63.1
B-Y11	100.0	4989.8	1060.7	100	4989.8	52.5	76.0	4386.7	1803.1	**76**	**4386.6**	69.1
B-Y12	35.5	2181.1	1342.4	35	2167.8	59.1	34.0	2173.6	1762.0	**34**	**2151.3**	60.8
B-Y13	149.0	7341.9	320.6	149	7341.9	51.6	109.0	6043.4	1811.7	**109**	**6022.4**	70.5
B-Y14	49.3	2951.6	1818.9	49	2915.4	63.7	48.0	2942.1	1802.8	**48**	**2888.3**	65.0
B-Y15	150.0	7360.5	320.6	150	7360.5	52.1	117.6	6477.7	1804.6	**117**	**6451.1**	66.7
B-Y16	53.4	3081.0	1594.5	53	3049.4	58.5	51.9	3089.8	1804.1	**51**	**3177.8**	60.8
B-Y17	198.0	9558.3	1098.2	198	9558.3	51.8	148.0	8462.3	1808.1	**148**	**8456.4**	69.3
B-Y18	65.3	3689.1	1969.6	**64**	**3665.6**	63.0	65.6	3676.7	1810.8	65	3647.1	62.1
B-Y19	199.0	9607.8	1106.2	199	9607.8	51.3	154.0	9298.8	1808.9	**154**	**9296.4**	66.2
B-Y20	75.2	4190.0	1875.9	74	4156.4	59.6	74.6	4298.5	1805.6	**73**	**4351.4**	60.4
Total	1547.5	3980.0	839.8	1542	3967.3	56.0	1253.9	3639.2	1234.5	1249	3629.6	65.5
Total-50				1146	5591.4	51.5				860	4895.7	69.0
Total-20				396	2343.2	60.6				389	2363.5	62.0

and 0.5% of the travel distance is saved by optional splits, while the mean run time increases to 300 second per instance.

The algorithm is clearly not suited for strongly heterogeneous box sets – and this comes not as a surprise. The packing approach is strictly tailored to weakly heterogeneous box sets with only few types of boxes. Furthermore, it makes use of box types with similar horizontal dimensions. Under these conditions a packing approach based on the extensive use of stacks and walls consisting of stacks makes sense. By such procedure simple and clear packing patterns are guaranteed that can be implemented in a comfortable fashion and needed running times can be held low. But these conditions are not given with the instances by Gendreau et al. (2006) where each box has specific dimensions and filling gaps in patterns is made further difficult by multiple additional constraints. All in all, algorithm SDVRLH2 is strictly limited to weakly heterogeneous box sets.

6. Conclusion

In this work, we deal with the vehicle routing problem with split delivery and three-dimensional loading constraints (3L-SDVRP) which is similar to the 3L-CVRP formulation by Gendreau et al. (2006) except for the only-one-visit assumption. The context of our research is given by problems of inbound heuristic with weekly heterogeneous box sets. Two problem variants are introduced, namely the 3L-SDVRP with forced splits and optional splits, respectively. In the former variant, only indispensable splits are allowed, while splitting deliveries is totally free in the latter variant.

We propose an algorithm that is able to cope with both problem variants. The method follows the principle “Packing first, routing second”, thus the packing and routing task are solved in two strictly separate steps. In the packing step patterns for one or two customers are constructed that consist of vertical layers. This ensures simple packing plans which are easy to implement in practice. Forced splits are realized by direct trips to single customers and related packing patterns that a good portion of the loading space. Packing is carried out by the container loading GA by Bortfeldt and Gehring (2001) and some constructive heuristics. Routing is done by a local search based on a giant tour representation. Optional splits utilise the layer structure of customer packing patterns.

Our algorithm is applied to three different sets of instances with industrial as well as academic origin. Comparisons to existing 3L-SDVRP methods prove a good quality of the results, and they are calculated in relatively short running times. The results with optional splits are generally better than the results calculated only with forced splits. This is especially true for instances with higher box volume / vehicle volume ratio per customer, where still no direct tours are necessary though. Thus, it has been proven that by optional splits tours and travel distance can be saved also in the context of routing problems with three-dimensional loading constraints. In an additional experiment with the instances by Gendreau et al. (2006), it turned out that the algorithm is only applicable to weekly heterogeneous box sets.

Future work should enable the hybrid algorithm to cope with further practical constraints related to containers, items, cargo and load (Bortfeldt and Wäscher, 2013; Pollaris et al., 2015), or related to operation and time, for example the loading priorities, time-windows or load bearing strength constraint.

Acknowledgments

Financial support from the NSFC research grant 71371162 and the Fujian Fumin Foundation is gratefully acknowledged. We also like to express sincere thanks to the three anonymous reviewers for helpful comments.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2019.09.024.

References

Archetti, C., Bianchessi, N., & Speranza, M. G. (2011). A column generation approach for the split delivery vehicle routing problem. *Networks, 58*, 241–254.
Archetti, C., Bianchessi, N., & Speranza, M. G. (2014). Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research, 238*(3), 685–698.
Archetti, C., Savelsbergh, M. W. P., & Speranza, M. G. (2008). To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review, 44*, 114–123.
Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research, 19*(1–2), 3–22.
Archetti, C., Speranza, M. G., & Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation science, 40*(1), 64–73.
Bartók, T., & Imre, C. (2011). Pickup and delivery vehicle routing with multidimensional loading constraints. *Acta Cybernetica, 20*, 17–33.
Belenguer, J. M., Martinez, M. C., & Mota, E. (2000). A lower bound for the split delivery vehicle routing problem. *Operations Research, 48*, 801–810.
Berbotto, L., García, S., & Nogales, F. J. (2014). A randomized granular tabu search heuristic for the split delivery vehicle routing problem. *Annals of Operations Research, 222*(1), 153–173.
Bianchessi, N., & Irnich, S. (2019). Branch-and-cut for the split-delivery vehicle routing problem with time-windows. *Transportation Science, 53*(2), 442–462.
Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega, 23*(4), 377–390.
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., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research, 131*(1), 143–161.
Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading a state-of-the-art review. *European Journal of Operational Research, 229*(1), 1–20.
Bruck, B. P., & Iori, M. (2017). Non-Elementary formulations for single vehicle routing problems with pickups and deliveries. *Operations Research, 65*(9), 1597–1614.
Ceschia, S., Schaerf, A., & Stützle, T. (2013). Local search techniques for a routing packing problem. *Computers & Industrial Engineering, 66*, 89–105.
Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), *Combinatorial optimization* (pp. 315–338). Chichester: John Wiley & Sons Ltd.
Côté, J. F., Guastaroba, G., & Speranza, M. G. (2017). The value of integrating loading and routing. *European Journal of Operational Research, 2571*(1), 751–759.
Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics, 50*(3), 239–254.
Dror, M., & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science, 23*(2), 141–145.
Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL), 37*(3), 383–402.
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.
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.
Helsgaun, K. (2009). General k-opt submoves for the lin–kernighan tsp heuristic. *Mathematical Programming Computation, 1*(2-3).
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.
Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *Top, 18*(1), 4–27.
Iori, M., & Martello, S. (2013). An annotated bibliography of combined routing and loading problems. *Yugoslav Journal of Operations Research, 23*(3), 311–326.
Irnich, S., Schneider, M., & Vigo, D. (2014). Chapter 9: four variants of the vehicle routing problem. *Vehicle Routing: Problems, Methods, and Applications* (2nd Ed., pp. 241–271). Society for Industrial and Applied Mathematics.
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.
Koch, H., Bortfeldt, A., & Wäscher, G. (2018). A hybrid algorithm for the vehicle routing problem with backhauls, time windows and three-dimensional loading constraints. *OR Spectrum, 40*, 1029–1075.
Li, X., Yuan, M., Chen, D., Yao, J., & Zeng, J. (2018). A data-driven three-layer algorithm for split delivery vehicle routing problem with 3D container loading constraint. In Proceedings of the KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom* (pp. 528–536). ACM. August 19–23.
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.
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.
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, 46*(1), 63–82.
Moreno, L., Poggi de Aragão, M., & Uchoa, E. (2010). Improved lower bounds for the split delivery vehicle routing problem. *Operations Research Letters, 38*, 302–306.
Moura, A. (2019). A model-based heuristic to the vehicle routing and loading problem. *International Transactions in Operational Research, 26*(3), 888–907.
Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *OR Spectrum, 31*(4), 775–800.
Ozbaygin, G., Karasan, O., & Yaman, H. (2018). New exact solution approaches for the split delivery vehicle routing problem. *EURO Journal on Computational Optimization, 6*(1), 85–115.
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.
Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2015). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum, 37*(2), 297–330.
Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2017). Iterated local search for the capacitated vehicle routing problem with sequence: Based pallet loading and axle weight constraints. *Networks, 69*(3), 304–316.
Qiu, M., Fu, Z., Eglese, R. W., & Tang, Q. (2018). A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups. *Computers & Operations Research, 100*, 102–116.
Rajappa, G. P., Wilck, J. H., & Bell, J. E. (2016). An ant colony optimization and hybrid metaheuristics algorithm to solve the split delivery vehicle routing problem. *International Journal of Applied Industrial Engineering, 3*(1), 55–73.
Reil, S., Bortfeldt, A., & Mönch, L. (2018). Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints. *European Journal of Operational Research, 266*(3), 877–894.
Shi, J., Zhang, J., Wang, K., & Fang, X. (2018). Particle swarm optimization for split delivery vehicle routing problem. *Asia-Pacific Journal of Operational Research, 35*(2), 1840006.
Song, X., Jones, D., Asgari, N., et al. (2019). Multi-objective vehicle routing and loading with time window constraints: A real-life application. *Annals of Operations Research*. https://doi.org/10.1007/s10479-019-03205-2.
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, 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-Mejia, C. A., Montoya-Torres, J. R., & Islam, S. M. N. (2019). A nonlinear optimization model for the balanced vehicle routing problem with loading constraints. *International Transactions in Operational Research, 26*(3), 794–835.
Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research, 234*(3), 658–673.
Wei, L., Zhang, Z., & Lim, A. (2014). An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *IEEE Computational Intelligence Magazine, 9*(4), 18–30.
Wilck, J. H., & Cavalier, T. M. (2012). A genetic algorithm for the split delivery vehicle routing problem. *American Journal of Operations Research, 2*, 207–216.
Wisniewski, M., Ritt, M., & Buriol, L. S. (2011). A tabu algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. In

Proceedings of the Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional (pp. 1502–1511).

Yi, J., & Bortfeldt, A. (2018). The capacitated vehicle routing problem with three-dimensional loading constraints and split delivery– A Case study. In *Proceedings of the Operations Research* (pp. 351–356). Springer.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2012). The pallet-packing vehicle routing problem. *Transportation Science, 46*(3), 341–358.

Zhang, D., Cai, S., Ye, F., Si, Y. W., & Nguyen, T. T. (2017). A hybrid algorithm for a vehicle routing problem with realistic constraints. *Information Sciences, 394/395*, 167–182.

Zhu, W., Qin, H., Lim, A., & Wang, L. (2012). A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers & Operations Research, 39*(9), 2178–2195.