A Multi-Compartment Vehicle Routing Problem with Loading and Unloading Costs

Alexander Hübner,[a] **Manuel Ostermeier**[b]

[a] Luxembourg Centre for Logistics and Supply Chain Management, University of Luxembourg, 1511 Luxembourg;
[b] Operations Management, Catholic University Eichstätt–Ingolstadt, 85049 Ingolstadt, Germany

Contact: alexander.huebner@uni.lu, http://orcid.org/0000-0002-1358-1284 (AH); manuel.ostermeier@ku.de (MO)

Abstract. This paper discusses a multi-compartment vehicle routing problem (MCVRP) that occurs in the context of grocery distribution. Different temperature-specific product segments (e.g., frozen or ambient) are transported from a retail warehouse to outlets. Different temperature-specific product segments can be transported together using multi-compartment vehicles. These trucks are technically able to have different temperature zones on the same truck by separating the capacity of a vehicle flexibly into a limited number of compartments. On one hand, this leads to a cost saving as different product segments ordered by one outlet can be delivered jointly using only one truck. This impacts the routing and the number of stops—i.e., the transportation costs and unloading costs. On the other hand, more than one shipping gate has to be approached at the warehouse to collect and load different product segments. As a consequence, the number of segments on each truck and therefore the number of compartments impact loading, transportation, and unloading costs.

An extended MCVRP with flexible compartments is presented to account for these loading and unloading costs. To solve the problem that arises, a large neighborhood search (LNS) tailored to the extended model is defined. The LNS includes problem-specific extensions in terms of the removal and reinsert operators as well as the termination criteria. It is tested using a case study with a retailer, benchmark data, and randomly generated data. Results are also compared to existing approaches. In line with the analyses performed for the model introduced, it is shown that the integration of loading and unloading costs into the model impacts routing considerably, and ultimately results in significant savings potential for retailers.

Funding: The research of Manuel Ostermeier is supported by the German Ministry of Education and Research and the Hanns Seidel Foundation.

1. Introduction

The transportation problem that will be discussed in this paper is a variant of the classic capacitated vehicle routing problem (CVRP) (for reviews, see Laporte 2009 and Toth and Vigo 2014) and occurs in the context of food distribution. Distribution is a major cost component of the grocery supply chain and amounts to about 20% of total logistics costs (Kuhn and Sternbeck 2013; Hübner, Kuhn, and Sternbeck 2013a). Given the narrow margins in grocery retailing, which are often less than 2%, efficient transportation planning is essential (Hübner, Kuhn, and Sternbeck 2013b; Agrawal and Smith 2015).

Groceries are stored in and transported from warehouses to the outlets in different temperature zones (e.g., deep-frozen, cold, ambient). The specific temperature requirements during transportation need to adhere to legislative regulations. For instance, in Europe, the mandated temperatures are −20°C to −18°C for deep-frozen products, +2°C to +7°C for cooled products (like meat and dairy products), and +4°C to +7°C for fruits and vegetables. Only ambient products like dry goods and beverages have no strict transportation temperature requirements. As a consequence, there are at least four temperature zones during transport, plus possibly stricter temperature controls required by retailers. An example of this is the transport of cooled and fresh products at even colder temperatures and in separate zones to obtain a longer shelf life (Akkerman, Farahani, and Grunow 2010; Amorim et al. 2013).

In the past, trucks have usually been customized for exactly one product segment with the corresponding temperature due to the specific temperature requirements. Recent truck models are equipped with temperature-specific compartments that allow the transport of different product segments in the associated chambers (compartments) of one truck. When considering deep-frozen and ambient products ordered by the same outlet, use of multi-compartment vehicles (MCVs) makes it possible to deliver both product segments on the same truck at the same time.

On the grounds of temperature requirements, retailers store, pick, and prepare the deliveries in temperature-specific warehouse sections. The use of MCVs therefore implies the need to pick up deliveries at different temperature-specific gates. At the same time, however, this enables the distribution of multiple segments within one truck. For example, Lekkerland received the German Logistics Award in 2013 for the introduction of multi-temperature logistics. For the retailer, it no longer seemed customer friendly, cost-efficient, or sustainable to supply customers with different single-compartment vehicles (SCVs) for each separate segment. With the award, Lekkerland was recognized as achieving bundling effects in operational processes, especially for customers (Grünrock-Kern 2013). Further retailers are increasingly investing in MCVs, especially for urban deliveries with small outlets and outlets with restricted access. Here, the combined delivery of different product groups is often required.

This gives rise to questions about the associated operations and logistics costs when using MCVs, and the design of vehicle routing given the opportunity to combine different product segment flows in one truck with flexible compartment sizes. The problem that arises can be classified as a multi-compartment vehicle routing problem (MCVRP). Supplementary to existing literature concerning MCVRP, the applied research introduces the following characteristics:

- The assignment of product segments to compartments determines the vehicle configuration for a specific tour—i.e., the number of different chambers on each truck.
- The number and size of compartments are flexible—i.e., compartment setup has to be determined for each MCV. The sizes of the activated compartments can be adjusted exactly to suit the transportation units so that there is no loss in capacity by using compartments.
- Cost factors are introduced to account for different processes related to compartment setups—i.e., the number of compartments determines operations and logistics costs.

Besides the planning of delivery tours, the MCVRP considers (i) demands for multiple inhomogeneous product segments and (ii) vehicles consisting of multiple compartments. It has to be determined (iii) how many compartments are set up on each vehicle on its tour and (iv) which customer orders from different product segments are combined on each tour.

The problem is NP-hard since it is a generalization of the CVRP (see, for example, Toth and Vigo 2014). A heuristic, namely a large neighborhood search (LNS), therefore was developed to fit the requirements of the extended model. To our knowledge, this is the first comprehensive model and solution approach that has so far been proposed for this problem. We will further analyze the economic impacts that stem from the introduction of MCVs, taking an end-to-end perspective on logistics cost.

The outline of the paper is as follows. Section 2 describes the problem context and identifies the decision-relevant costs. Section 3 discusses the related literature. Next, the mathematical formulation and formal definition of the problem are introduced in Section 4. The LNS developed is presented in Section 5. In Section 6, our solution approach is applied to numerical examples, first using randomly generated instances to examine the performance of the implementation, and afterwards with real-life data from a case study. Finally, the findings and conclusions are summarized in Section 7.

2. Problem Description

This section will lay the foundations for the decision model (Section 4) by defining the typical network structures and the impact of MCVs and by identifying the decision-relevant costs as well as technical considerations when using MCVs. This in turn enables us to define the underlying decision problem. All information has been gathered in close collaboration with a retailer.

Characteristics of grocery network structures. Grocery retailers channel the overwhelming majority of product volumes to their stores via distribution centers (DCs) (Fernie and Sparks 2009; Kuhn and Sternbeck 2013). European discounters and most full-line supermarkets operate their own distribution networks consisting of several regional DCs from which the complete assortment is supplied. DCs are organized according to temperature-specific product segments. A retail DC usually serves between 50 and 400 outlets (Glatzel, Großpietsch, and Hübner 2012).

Impact of MCVs on distribution system and costs. Using MCVs increases the flexibility for the delivery of products but also poses some new challenges with regard to planning. Figures 1 and 2 visualize the differences between the use of SCVs and MCVs. Figure 1 shows a classical distribution system with SCVs, where a tour starts with a stop at exactly one temperature-specific shipping gate of the DC. Having loaded all products of one segment (i.e., goods from an identical temperature zone) for a tour, outlets are supplied with one delivery each. If there are orders for another product segment with different temperature requirements, a second truck is needed. The additional truck only

Figure 1. (Color online) Distribution System with Single-Compartment Vehicles

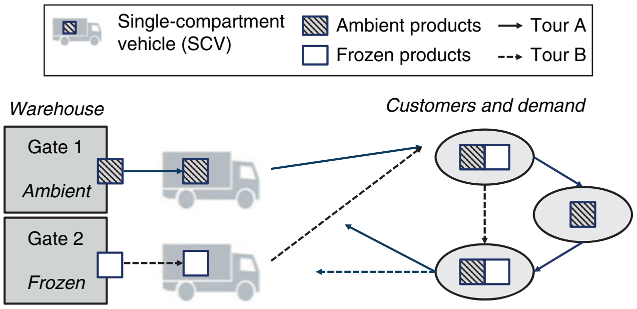
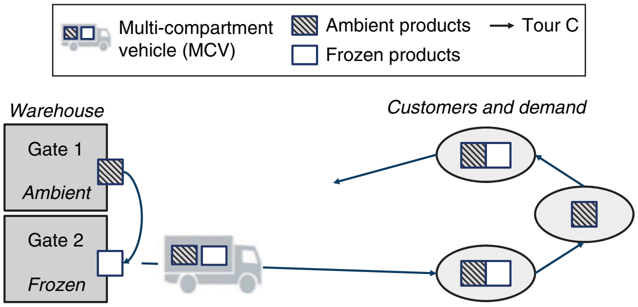


Figure 2. (Color online) Distribution System with Multi-Compartment Vehicles



serves customers who have ordered the second product segment. Another delivery route therefore has to be set up (see Tour A and B in Figure 1).

The distribution system with MCVs involves different processes (see Figure 2). Here, the MCV approaches both shipping gates at the DC to collect goods with differing temperature requirements. Having loaded all product segments, the MCV is able to supply outlets with multiple product segments on the same tour. The decision-relevant times and processes needed for MCV-specific operations have been assessed in a time and motion study with a retailer. Accordingly, there are two major process and cost differences between these two different distribution systems.

Type (1): MCVs require additional operational processing time. The pick-up of different product segments at the DC results in additional stops at the DC. This leads to additional travel and setup time for loading. The setup time includes traveling to each gate, rearranging and opening the lift gate, and the specific compartment walls. The loading costs therefore are setup costs that depend on the number of compartments as this represents the number of setups at different shipping gates.

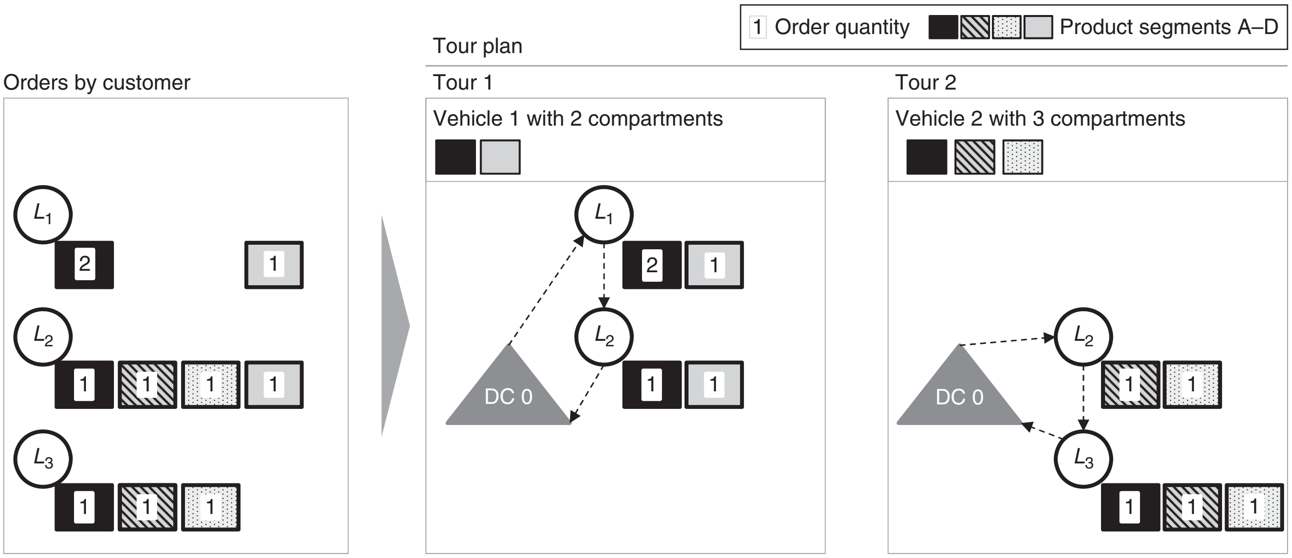
Type (2): MCVs reduce operational processing time. Being able to transport multiple product segments jointly offers the advantage of serving customers with more than one product segment at one stop. This means that the number of stops and setups for unloading at the retail outlet can be reduced.

Going into more detail, unloading costs are setup costs that occur for each stop of the vehicle at a retail store and can be split into (a) general setup costs and (b) compartment-specific setup costs. (a) General setup costs per stop (e.g., backward drive to ramp, opening of truck) are independent of the actual vehicle-compartment setting as they apply for processes that are equal for both SCVs and MCVs. These costs depend only on the number of stops (i.e., the number of customers on a tour) and include the vast majority of all decision-relevant unloading processes. (b) Compartment-specific setup costs occur only for the opening of compartment doors. These costs therefore depend on the actual number of compartments used at each stop (i.e., the number of compartments to be opened and closed at each stop). Compared to costs (a), costs (b) are very minor costs as compartment operations take only a few seconds (i.e., the closing of the door). Moreover, they are almost the same if multiple compartments are active. For these reasons, it is only necessary for the unloading costs to differentiate between the use of SCVs and MCVs, and not the actual number of compartments. In other words, costs (b) only differ significantly if one or at least two compartments are used. As a consequence, one does not need to differentiate unloading costs by customer (e.g., if one customer receives products in two compartments and the other customer receives products in three compartments). Also, the higher flexibility with MCVs for building vehicle tours will reduce driving distances compared to SCVs.

The different cost types (1) and (2) evoke a tradeoff for the delivery of different product segments, which needs to be considered in the VRP, particularly when selecting the number of compartments per tour. On one hand, adding further product segments to one vehicle and thus activating additional compartments results in higher loading costs. On the other hand, unloading costs are reduced because of the joint delivery of different segments with one tour. Note that the actual loading and unloading—i.e., moving roll-cages in and out of the truck—is required regardless of the number of compartments and is not decision relevant in this case. Of the total costs consisting of transportation, loading, and unloading costs, the latter two often contribute up to 20% of the total costs (see also Section 6).

Technical configuration of MCVs. Each MCV can be customized showing different compartment settings. This means that the number of compartments, their size, and consequently, their capacity is not predefined for each tour. Compartments can also not be used (i.e., deactivated) by removing the partitioning wall. The smallest possible size of an active compartment is one transportation unit (TU; e.g., one roll-cage or standard

Figure 3. Illustrative Example of an MCVRP with a Flexible Number of Compartments and Flexible Compartment Sizes



pallet) so that there is no loss in transportation capacity due to the use of compartments. For instance, if the vehicle capacity is 33 standard pallets, the compartment sizes might be adjusted for each compartment to the size of $0, 1, 2, \ldots, 33$ TUs, as long as the total vehicle capacity is not exceeded. The latest generation of MCV can be loaded from the back using a longitudinal separation.

Decision problems to create tour plans with MCVs. After having identified the impact on processes and the distribution system, we would like to illustrate the corresponding decision problem. Figure 3 displays the decision problems in grocery retailing for MCVRP supplying three customers with four product segments. Customer locations L_1 to L_3 are depicted as circles and the product segments A to D as squares. The number within a square indicates the order quantity of this customer for this particular product segment. In the example, the order size for various product segments varies between customers, so that, for example, customer 1 does not order segments B and C, and customer 3 has no demand for segment D. Furthermore, each vehicle has a capacity of five units, and the maximum number of compartments equals the maximum number of product segments.

A possible tour plan for this particular problem is shown on the right of Figure 3. Creating the tour plan requires solutions for the product segment vehicle assignment, the number of compartments per vehicle, the order vehicle assignment, and the location sequencing. In this example, two vehicles are required to serve the customers. If a certain product segment is assigned to a vehicle, the associated square in the top left part of the box is shown accordingly. Furthermore, if a certain order is assigned to a vehicle, the corresponding product segment square with the order size is shown below the corresponding location. Finally, the vehicle routes are indicated at the edges. It also shows that all compartment types can be combined in one vehicle (e.g., compartment for product segments A and D in tour 1, and compartment for product segments B and C in tour 2), and that there is no predefined sequence when activating compartments.

Summary. The VRP for grocery distribution with MCVs with flexible compartments needs to include further cost components to reflect the operational processes. MCVs induce higher loading and stop costs at the DC as multiple shipping gates may need to be approached by one truck, lower unloading and stop costs at the retail outlets, and potentially lower transportation distances as customers may be supplied with one truck that jointly transports different product segments. This tradeoff between time saving for combining orders and the expenses for additional loading actions is another decision that needs to be evaluated for MCVRP.

3. Related Literature

While there is a wide range of publications dealing with the VRP and its various extensions (see Golden, Raghavan, and Wasil 2008 and Toth and Vigo 2014), there is only a small field of research for problems concerning deliveries with MCVs (see Pollaris et al. 2015).

The literature on MCVRPs can be divided into two streams with (1) fixed or (2) flexible compartments. As we consider an MCVRP with flexible compartments, we give a short overview of publications in (1) before moving on to the relevant literature in (2).

Table 1. Related Literature on MCVRP with Flexible Compartments

Authors	Application	Costs	Solution	Extension/comment
Chajakis and Guignard (2003)	Food	TC	SH	Including boxes and cooling costs
Derigs et al. (2011)	Food and petrol	TC	SH	Test of various algorithms
Pirkwieser, Raidl, and Gottlieb (2012)	Food and petrol	TC	VNS	Load building
Henke, Speranza, and Wäscher (2015)	Waste	TC	VNS	Discrete compartment sizes

Note. SH, Specialized heuristic; TC, transportation costs; VNS, variable neighborhood search.

(1) The first stream on *fixed compartments* deals mainly with problems in the context of petrol distribution. For instance, Brown and Graves (1981) develop an automated real-time dispatch system for the distribution of petroleum products. Avella, Boccia, and Sforza (2004) apply a branch-and-price algorithm for the supply of fuel pumps. Other applications are, for example, given by Lahyani et al. (2015), who published a case study on the collection of olive oil of differing qualities that is based on a branch-and-cut algorithm. El Fallahi, Prins, and Wolfler Calvo (2008) study the performance of a memetic algorithm as well as a tabu search on classical VRP instances extended by the use of compartments. Muyldermans and Pang (2010) compare the use of MCVs in the area of waste collection. Additionally, they analyze the improvement seen when introducing MCVs compared to the use of SCVs. In their approach, a guided local search procedure is used to solve specific routing problems. A memetic algorithm is also used by Mendoza et al. (2010) to solve an MCVRP with stochastic demands. Mendoza et al. (2011) further develop construction heuristics for their approach. This involves one savings-based algorithm and two new and more foresighted approaches that focus first on routing and in a second step on clustering. Melechovský (2013) extends an MCVRP by considering time windows and fleet size, solving the resulting problems with a variable neighborhood search (VNS).

(2) Apart from the aforementioned publications on fixed compartment sizes, there is only limited literature available for MCVRP with *flexible compartments*. Table 1 structures the related literature.

Chajakis and Guignard (2003) develop a model for delivery of groceries and use a heuristic based on Lagrangian relaxations. In addition to transportation costs, they also consider cooling costs for the corresponding compartments or boxes as decision relevant, but no cost differentiation for the used vehicle configuration is made. Moreover, they only focus on the order vehicle assignment of the VRP and use an approximation of Fisher and Jaikumar (1981) for transportation costs. The maximum problem size examined is restricted to seven vehicles and therefore only small instances are being researched.

Derigs et al. (2011) are the first to present a comprehensive problem formulation for an MCVRP that minimizes transportation distances. Their model is formulated to address both the industry of food and petrol distribution. Derigs et al. (2011) provide a solver suite to tackle MCVRPs. Different algorithms successfully used on various VRP settings compound this solver suite. It consists of construction heuristics (e.g., savings) and improvement heuristics such as LNS, local search, and metaheuristics. The different solution approaches are applied to a set of benchmark instances. As a result, they identify a heuristic approach combining multiple well-known algorithms for MCVRPs. Based on Derigs et al. (2011), Pirkwieser, Raidl, and Gottlieb (2012) introduced a VNS for the MCVRP especially focusing on the packaging aspect. This involves a measure to distinguish packaging, leading to solutions with a higher packing density. Additionally, a new neighborhood structure is proposed to enhance the search for improvements.

Similarly, Henke, Speranza, and Wäscher (2015) consider an MCVRP for the collection of glass waste with flexible compartments that can only be discretely varied—i.e., there are predefined possible sizes for each compartment. For instance, if we assume a total capacity of 100 TUs, the size of a single compartment might be 50, 25, or 15 TUs. They apply a VNS to solve the problem. In their work, they also consider the assignment of product types to compartments, which leads to different vehicle settings. The case where fewer compartments than product types are available is especially considered. However, in contrast to the MCVRP with loading and unloading costs, the compartment setting is only influenced by the associated delivery distances and not by the choice of compartments.

Summary. In all of the papers presented, the cost of transportation is the only cost driver that has been taken into account. No differentiation was made between vehicles with one or more compartments and hence the related costs for each additional compartment used. Consequently, no operational processes were considered that depend on the number of compartments. In most cases, only randomly generated problem settings have been considered, and there is a lack of applications with empirical data to date. We consider Derigs et al. (2011) as a benchmark (for the special case without loading/unloading costs) as they regard a similar problem setting and especially as they formulate a model for the use of completely flexible compartments.

4. Model for the MCVRP with Loading and Unloading Costs

This section develops the model for minimizing loading, transportation, and unloading costs related to vehicle routing with MCVs—that is, finding the cost-optimal tours by assigning orders of specific product segments to vehicles and compartments, and building optimal delivery tours. We formulate the MCVRP_LU, which is distinguished by an objective function accounting for **L**oading and **U**nloading costs depending on the differing number of compartments per vehicle.

The MCVRP_LU can be formulated as follows. Let $G = (L, E)$ be an undirected, weighted graph consisting of a vertex set $L = \{0, 1, \ldots, n\}$, representing the location of the DC ($\{0\}$) and the locations of n customers *cust* ($L_{cust} = \{1, \ldots, n\}$), and a set of edges $E = \{(i, j): i, j \in L, i < j\}$, representing the connection between different locations. A nonnegative distance cost $cost_{ij} \in E$ is assigned to each of these edges. It is assumed that all distances satisfy the triangle inequality and each tour starts and ends at the DC.

Orders are defined by customers, product segments, and quantity. The set of orders is denoted by O and the set of product segments by P. In our case, a product segment consists of products that belong to one temperature zone and hence products that can be transported jointly within one compartment. Each customer $j = 1, \ldots, n$ may place one or several orders, each referring to a single product segment. For each order $o \in O$, $customer(o) \in L_{cust}$ denotes the customer that placed the order, $product(o) \in P$ the product segment ordered, and $quantity(o) > 0$ the quantity ordered, so that for each order a positive demand exists. The orders have to be collected from the DC and transported to the customers. A customer may be visited several times (i.e., during different tours) to deliver different product segments. Each customer, $j \in L_{cust}$, places at least one order, and there is at least one order for each product segment, $p \in P$. This also means that not all customers order from each segment. A split delivery of one product segment of a single customer is not possible.

For the purposes of transportation, a set of vehicles V is available, each equipped with the identical total capacity ($vehCapa > 0$) and a set of compartments denoted by C. The set of available vehicles is assumed to be sufficiently high to meet customer demand. Total vehicle capacity *vehCapa* can be divided into a limited number $\bar{c}$ of compartments, $\bar{c} \leq |C|$, for each vehicle, $v \in V$. In the simplest case, the entire vehicle capacity is used for one compartment. If more than one compartment is used, the size of each compartment is determined by the order quantity assigned to it. This means the assignment of orders to one compartment influences the possibility to assign further product segments to other compartments as total capacity is limited by *vehCapa*. As a consequence, compartment sizes are flexible between 0—i.e., the compartment is not used on the vehicle—and the vehicle capacity. Any loss in capacity is eliminated because of the flexible adjustment and the use of different compartments. Furthermore, the number of product segments and the associated number of compartments on each vehicle is decision relevant. It has to be taken into account that different product segments cannot be mixed in the same compartment.

Finally, all orders assigned to a compartment on a vehicle have to be sequenced to determine the tour of the corresponding vehicle. Thus, a solution with $m \in$ tours is created, equivalent to the use of m vehicles. For instance, a tour k has the form $t_k = (0, stop_1, \ldots, stop_l, 0)$, with 0 as the DC and $stop_j \in L_{cust}$, $j \in \{1, \ldots, l\}$.

The model developed minimizes the sum of loading, transportation, and unloading costs by selecting the number of compartments per vehicle, assigning product segment–specific orders to compartments, and creating tour plans such that vehicle capacities are met. As a result, the following decisions have to be taken simultaneously.

- *Sequencing delivery tours* expressed by binary variable b_{ijv}, indicating whether vehicle v travels from location i to j, and integer variable u_{iv}, determining the position of customer i on the tour of vehicle v ($u_{iv} \in \{0, \ldots, |L|\}$);
- *Assigning orders to vehicles and compartments* expressed by binary variable x_{ovc}, indicating whether order o is delivered by vehicle v in compartment c. This decision includes which customer orders will be combined and therefore which locations have to be visited jointly;
- *Determining the number of compartments on each vehicle* denoted by the binary auxiliary variable a_{vc}, which is set equal to one, if compartment $c \in C$ is used for transportation. Note that this decision is already included in x_{ovc}, as $\sum_{o \in O} x_{ovc} \leq |O| \cdot a_{vc}$; the associated number of active compartments d_v of each vehicle v is defined accordingly by $d_v = \sum_{c \in C} a_{vc}$.

Note that the sequencing and order assignment decisions are taken in each VRP, while the assignment to compartments and selection of the number of compartments define the uniqueness of the MCVRP_LU. As the evaluation of loading/unloading processes is a central aspect of the MCVRP_LU, we further introduce loading costs lc_{d_v} and unloading costs ulc_{d_v}, both depending on the number of compartments d_v used on each vehicle, $v \in V$. More specifically, *lc* and *ulc* can be regarded as cost vectors representing costs for each vehicle, and their value is determined by the vehicle setting—i.e., the number of active compartments.

The objective function and constraints of the model are formulated as follows.

min! Total Costs

$$= \sum_{v \in V} lc_{d_v} + \sum_{i \in L} \sum_{j \in L} cost_{ij} \cdot b_{ijv} + ulc_{d_v} \sum_{i \in L} \sum_{j \in L_{cust}} b_{ijv} \tag{1}$$

subject to

$$\sum_{j \in L_{cust}} b_{0jv} \leq 1 \quad v \in V, \tag{2}$$

$$\sum_{i \in L} b_{ihv} = \sum_{j \in L} b_{hjv} \quad v \in V, h \in L, \tag{3}$$

$$u_{iv} - u_{jv} + |L| \cdot b_{ijv} \leq |L_{cust}| \quad v \in V, i \in L, j \in L_{cust}, \tag{4}$$

$$u_{0v} = 1 \quad v \in V, \tag{5}$$

$$\sum_{o \in O} \sum_{c \in C} quantity(o) \cdot x_{ovc} \leq vehCapa \quad v \in V, \tag{6}$$

$$\sum_{v \in V} \sum_{c \in C} x_{ovc} = 1 \quad o \in O, \tag{7}$$

$$\sum_{o \in ordCust(j)} \sum_{c \in C} x_{ovc} \leq |O| \cdot \sum_{i \in L} b_{ijv} \quad v \in V, \ j \in L_{cust}, \tag{8}$$

$$\sum_{o \in O} x_{ovc} \leq |O| \cdot a_{vc} \quad c \in C, v \in V, \tag{9}$$

$$\sum_{o \in O_p} x_{ovc} \leq |O| \cdot (1 - x_{rvc}) \quad v \in V, c \in C, \\ p, q \in P \colon p \neq q, r \in O_q, \tag{10}$$

$$a_{vc} \in \{0, 1\} \quad v \in V, c \in C, \tag{11}$$

$$b_{ijv} \in \{0, 1\} \quad i, j \in L, v \in V, \tag{12}$$

$$u_{iv} \in \{1, \ldots, |L|\} \quad i \in L, v \in V, \tag{13}$$

$$x_{ovc} \in \{0, 1\} \quad o \in O, v \in V, c \in C. \tag{14}$$

The objective function (1) of the MCVRP_LU minimizes the total costs across all vehicles v. The first term considers total loading costs, which depend on the number of active compartments d_v on each vehicle and therefore on the number of segments delivered within each tour. Note that the loading costs represent setup costs for the approach of different shipping gates. These costs occur for each vehicle when an additional compartment and product segment is added to the tour (see Section 2). In the second term, the total transportation costs are calculated. They are represented by the distance costs $cost_{ij}$ between locations i and j, and the chosen travel sequence from location i to j of vehicle v, indicated by the binary variable b_{ijv}. Finally, the third term represents the total costs for unloading at all customers supplied. The total costs for unloading consist of the vehicle-compartment-dependent costs ulc_{d_v} multiplied by the number of stops on the corresponding tour. Since the start and end of each tour are not counted as stops, only set L_{cust} has to be considered for the second sum. The unloading costs differ between vehicles with one and more compartments (see Section 2), also represented by the number of compartments d_v on each vehicle. It is therefore not necessary to differentiate the unloading costs by the number of compartments needed for each customer. It is important to note that because of the subscript that involves a decision variable and the last term of the objective function, the objective function presented is nonlinear.

Constraints (2) and (3) ensure that every vehicle v can only depart once from the DC ($i = 0$) and that every vehicle that arrives at a customer location j also departs from there. Restrictions (4) and (5) are used to eliminate subtours by indicating the position of customer i on the tour of vehicle v and setting the DC as the start and end point of each tour. This is imposed by the fact that the position of customer j is higher than that of i, if the vehicle v travels from i to j. Constraints (6) ensure that the orders loaded into all compartments of vehicle v do not exceed the vehicle capacity *vehCapa*. Each order o can only be assigned to one compartment c on a vehicle v, and therefore Equations (7) are needed. With constraints (8) it is ensured that customer j has to be visited if an order o of customer j is loaded on vehicle v. Here, the orders of customer j are denoted by $ordCust(j)$: $\{o \in O \mid customer(o) = j\}$. Constraints (9) ensure that compartment c on vehicle v is set to active if an order o is assigned to it. The incompatibility restrictions (10) make sure that incompatible orders from different segments p and q, $p, q \in P$ are not assigned to the same compartment. This is done by ensuring that an order o from the set of orders O_p, which comprises all orders of product segment p, is not combined in one compartment with an order r from the set O_q, which comprises all orders from product segment q. Last, the domains for the decision variables are defined by (11)–(14).

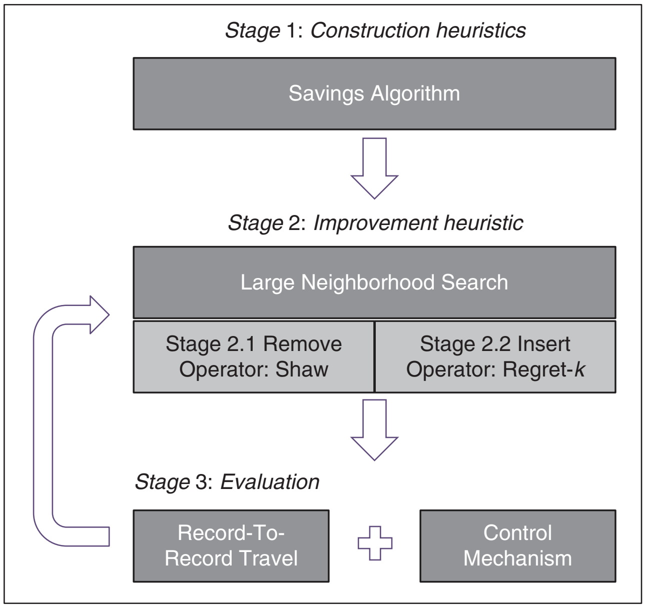
5. Solution Approach

As an extension of the CVRP, the MCVRP_LU is an NP-hard optimization problem, and one has to resort to heuristic methods to solve practical problem sizes of up to 400 store deliveries. For this, we choose the Savings algorithm by Clarke and Wright (1964) as the construction heuristic and an LNS as an improvement heuristic to solve the MCVRP_LU. At this point, we would like to note that in our case, the search within the LNS is based on the movement of orders between vehicles and not—as in classical VRP formulations—on the movement of customers. In the following, we outline the reasons for our choice of LNS to solve our problem and give a detailed description of the approach.

5.1. Heuristic Selection and Motivation

We build our solution approach on the investigations of Derigs et al. (2011). They tested combinations of different heuristics and found that the LNS shows the best performance for MCVRP. They combine several remove and insertion operators for their LNS approach and demonstrate that this can obtain good solutions

Figure 4. Algorithmic Structure


Stage 1: *Construction heuristics*
Savings Algorithm
Stage 2: *Improvement heuristic*
Large Neighborhood Search
Stage 2.1 Remove Operator: Shaw
Stage 2.2 Insert Operator: Regret-*k*
Stage 3: *Evaluation*
Record-To-Record Travel
Control Mechanism

for the MCVRP (without loading and unloading costs). Looking at the operators implemented within the LNS, the Shaw Removal (Shaw 1997) has been identified as the most promising removal process. Among the insertion operators, a regret insertion showed to be essential. Inspired by these results, we chose to implement an LNS based on Shaw (1997) for the removal of orders and Ropke and Pisinger (2006) as a regret insertion. These operators have been adjusted to exactly fit the requirements of the MCVRP_LU. Figure 4 illustrates the algorithmic structure of our approach. Section 5.2 presents the procedures used within the solution approach.

5.2. Solution Steps

The individual steps of the selected heuristic approach are presented below. The operators within the LNS as well as the governing process constructed are explained in greater detail. Finally, the algorithm used is presented.

Stage* 1: *Construction Heuristic. The parallel Savings algorithm by Clarke and Wright (1964) is used to create an initial solution. It is a widely applied construction heuristic also for MCVRPs (e.g., Muyldermans and Pang 2010), and it is able to produce a fast feasible solution. First, one location route is built for each order. Second, the savings effect of combining any two routes is calculated by evaluating the change in distance. For that, the savings value $saving$, with $saving_{ij} = dist_{i,0} + dist_{0,j} - dist_{i,j}$, is calculated for every order, where 0 represents the DC and $dist_{i,j}$ denotes the distance between two customer locations i and j. A savings list is created after evaluating the reduction in distance for all pairs. Then, tours that adhere to the constraints are combined by connecting last and first orders starting with the highest saving and working down the ordered savings list. The parallel version has been chosen as it tends to provide better solutions applied to a CVRP than the sequential approach (Laporte 2009).

Figure 5. Pseudocode for Stage 2.1 Shaw Removal

```
Input: Number of orders for removal $r$
  Randomly choose one order $o$ for removal from all orders $O$
  while Number of removed orders $o$ < Number of orders for
      removal $r$ do
    Step 1: Randomly select one order $o$ among the already
      removed orders $O^-$
    Step 2: Rank all remaining orders $s \in O^+$ in descending order
      based on $R_{os}$ that is calculated with Equation (15)
    Step 3: Draw a random number $z \in [0,1)$ and remove the order $s$
      which lies $z^\alpha$ down the ranking
  end while
```

Stage* 2: *Large Neighborhood Search. The LNS is used to ruin and recreate large parts of the results already obtained to achieve a solution improvement. Our LNS is based on the model formulation given in Section 4, so we evaluate our results using Equation (1).

Stage* 2.1: *Shaw Removal. We chose the Shaw Removal presented by Shaw (1997) as the remove operator as it fits the requirements of the MCVRP_LU well since it enables us to consider orders with a similar structure. This is an essential characteristic of the search procedure as the combination of different product segments leads to an increase in total costs. Beyond that, the Shaw Removal showed the best performance of all removal procedures within the solver suite applied by Derigs et al. (2011).

The removal approach is based on a defined similarity measure R_{os} between any two orders o and s (either from the same customer or different customers) with $o, s \in O$ and a randomized selection. In total, a defined number of r orders has to be removed from all orders O. In the following, we divide the set of all orders O into removed orders (O^-) and assigned orders (O^+), such that $O^+, O^- \subseteq O$, $O^+ \cup O^- = O$, and $O^+ \cap O^- = \emptyset$. Note that all orders have already been assigned to a tour after the completion of the construction heuristic (Stage 1). The first order o removed is chosen randomly from all orders O. Further orders are gradually removed according to a defined procedure based on the calculated similarity measure R_{os}. The pseudocode of the removal process is displayed in Figure 5 and detailed in the following.

After one order o has been selected randomly in Step 1, the similarity of orders is calculated in Step 2. The similarity index for all orders is calculated using the measure R_{os} that is defined in Equation (15). It expresses the similarity of two orders o and s, with

$o \in O^-$ and $s \in O^+$. The higher the calculated value of R_{os}, the less similar the orders compared

$$\frac{1}{R_{os}} := \phi \cdot \frac{cost_{os}}{cost_{\max}} + \omega \cdot prod_{os} + \psi \cdot \frac{|quantity(o) - quantity(s)|}{quantity_{\max}}. \qquad (15)$$

The measure R_{os} involves three metrics for the similarity of two orders in terms of *distance cost*, *product segment*, and *order size*. Weights ϕ, ω, and ψ are applied to represent the importance of each of the three components.

The first metric is the *distance cost* between customer locations, which represents the costs related to transportation and unloading. A high similarity among orders means that in this case orders are from the same customer (and have a positive impact on the unloading and transportation costs) or from customers who are located in close proximity (and have a positive impact on transportation costs). The similarity of two orders in this dimension is expressed by the ratio of $cost_{os}$ and $cost_{\max}$. The first represents the distance costs between the location of the customer who placed order o and the location of the customer who placed order s. The latter represents the maximum distance cost between any two customer locations.

The *product segment* is the second metric in which $prod_{os}$ indicates whether the orders belong to the same product segment. This implies that the orders can be assigned to the same compartment. It is defined as

$$prod_{os} := \begin{cases} 1 & \text{if } segment(o) \neq segment(s), \\ 0 & otherwise. \end{cases} \qquad (16)$$

A combination of two orders (o, s), $o, s \in O$ that are compatible (i.e., that belong to the same segment) and can thus be assigned to the same compartment have a higher similarity (i.e., a lower value for R_{os}). The loading of more product segments results in higher loading efforts, so the loading costs incurred by inserting orders from different product segments are implicitly evaluated in the similarity measure R_{os}.

Finally, the *order size* represents the fact that swapping orders of the same size tends to provide feasible solutions more quickly. This is expressed by the order quantity difference between orders o and s, where $quantity(o)$ and $quantity(s)$ are the quantity ordered and $quantity$(max) is the highest order quantity over all orders.

In Step 2, the orders are ranked according to similarity in descending order—i.e., the most similar order at the top and the least similar order at the bottom. Then, the next order for removal needs to be selected in Step 3. This selection process is based on the random number $z \in [0, 1)$ and a parameter α. Thus, for the selection of a new order, it is not the order with the highest similarity that is chosen but one that can be found z^{α}% down the similarity ranking. If the resulting position is not integer, the result is rounded to the next integer value.

Figure 6. Pseudocode for Stage 2.2 Regret-k Insertion

```
Input: Degree of regret $k$
  while Not all removed orders $o$, $o \in O^-$, are reinserted to $O^+$ do
    Step 1: Calculate $regret_k$ for each removed order $o$
    Step 2: Insert order $o$ with highest value of $regret_k$
  end while
```

The parameter α can be seen as a parameter for diversification. If α is chosen to be 1, the similarity is not taken into account and the choice of orders is completely random. The higher the value of α is set, the more decisive is the calculated similarity. In other words, the choice of a high value for α corresponds to an intensification, while a low value provides a more diversified removal. After r orders are removed, the algorithm continues with Stage 2.2 and the reinsertion.

Stage 2.2: Regret-k Insertion. Following the approach by Ropke and Pisinger (2006), we reinsert removed orders o, $o \in O^-$ with a regret-k operator to the set O^+. The pseudocode is displayed in Figure 6 and detailed in the following.

We modified the regret insertion of Ropke and Pisinger (2006) by accounting for all costs considered in the MCVRP_LU. Apart from the transportation costs, this also includes the influence on loading and unloading costs if an order is assigned to a tour. In Step 1, the regret value $regret_k$ is calculated for each removed order $o \in O^-$ using Equation (17), with $k \geq 2$. It is the sum of differences between the best option (represented by the lowest total tour costs $totalcost_{v_1}$) and k-best options (represented by $totalcost_{v_u}$). Here, $totalcost_{v_u}$ are the total tour costs of the uth best tour v if order s is inserted there and $totalcost_{v_u} \leq totalcost_{v_t}$, $\forall\, u < t$

$$regret_k := \sum_{u=2}^{k} (totalcost_{v_u} - totalcost_{v_1}). \qquad (17)$$

In Step 2, the order o with the highest regret value is chosen to be inserted into the tour that is best suited to it. This means the insertion does not just consider the actual state but also uses the regret criterion to evaluate possible future costs.

After the order s with the highest regret is inserted into O^+, the regret value is recalculated for each order o on the removal list O^-, as the insertion options might have changed. The insertion procedure is thus iterated until all removed orders $o \in O^-$ are allocated to a new position in the tour planning, and hence $O^- = \oslash$ and $O^+ = O$.

Stage 3.1: Record-To-Record Travel. To govern the process of finding a solution improvement using the LNS, the Record-To-Record Travel (RRT) metaheuristic is used. It was shown that the RRT delivers good solutions for the MCVRP. The RRT is defined according to Dueck (1993) and controls the improvement steps by setting a limit for the acceptance of declining solution values. This means that a solution is only accepted if it lies within a defined deviation D from the best result found so far. Even though the RRT approach is quite simple, it does not seem to be inferior to other methods used (Derigs et al. 2011). The scheme of the algorithm is as follows.

1. Check if new solution $S' \leq$ best solution (record) S_{best} + allowed deviation D
2. If true: accept new solution S'
3. Else: keep old solution S_{best}
4. Additionally: If new solution $S' <$ best solution S_{best}, then $S_{best} := S'$

Stage 3.2: Control Mechanism. For the regulation of the application runtime, additional parameters are integrated compared to the approach applied by Derigs et al. (2011). If the number of unsuccessful iterations reaches a reset border, the next iteration of LNS is run with a high value (e.g., one half of the order list) for the number of orders removed (r) to create a completely different neighborhood, thus aiming to avoid local minima. A limit is set for the maximum number of succeeding fruitless iterations to terminate the algorithm.

Overview. The procedure of Stages 1–3 denoted above has been incorporated into the LNS, and the algorithmic structure can be seen within the complete LNS description in Algorithm 1. The solution representation includes all critical information for the built tours. This involves tour number, sequence of customers visited together with the corresponding orders, number of compartments, and capacity used.

Algorithm 1 (Large neighborhood search)

```
Input: (Initialize solution $S$, set remove parameter $r$,
        set regret parameter $k$)
  $S_{best} := S$
  Shaw Removal $SR(r)$ according to Stage 2.1
  Regret Insertion $RI(k)$ according to Stage 2.2
  Set allowed deviation according Record-To-Record
     Travel $RRT()$
  while improving = true do
     $S' := S$
     Remove $r$ orders from $S'$ using $SR(r)$
     Reinsert removed orders into $S'$ using $RI(k)$
     if $ObjectFunction(S') < ObjectFunction(S_{best})$ then
        $S_{best} = S'$
        $S = S'$
        Reset number of unsuccessful runs to zero
     else if $ObjectFunction(S') \leq ObjectFunction(S_{best})$
           plus accepted deviation $D$ then
        $S := S'$
        Increase number of unsuccessful runs
     else
        Increase number of unsuccessful runs
        Continue with original solution $S$
     end if
     if Number of unsuccessful runs = limit then
        Set improvement false
     else if Number of unsuccessful runs = reset
           border then
        Remove high number of orders from $S$ using
           $SR(r)$
        Reinsert removed orders into $S$ using $RI(k)$
     end if
  end while
  return $S_{best}$
```

Summary. As the solution of practically relevant problem sizes is a central point of our research, we solve the MCVRP_LU with an LNS as it has shown good results in other MCVRP instances and as we solve large problem sizes with up to 400 customers. To express the particularities of the MCVRP_LU with different product segments that can be transported jointly, our search within the LNS is based on the movement of customer- and segment-specific orders between vehicles. We decided to use the operators Shaw Removal and regret insertion. In contrast to Derigs et al. (2011), these are detached from other operators to formulate a leaner LNS version for the MCVRP that is also able to provide a good solution quality, as we will show in Section 6. We decided to use this leaner LNS version as we found that other remove operators could not provide a significant contribution to the search procedure. This is because of the particular problem structure with multiple orders from the same customer and also the direct relation of orders to product segments. The removal operator of Shaw (1997) has been adjusted to fit the requirements of the MCVRP_LU well by measuring similarities for transportation and loading/unloading costs, and specifically by introducing a product segment similarity. The regret insertion of Ropke and Pisinger (2006) is extended to take into account total tour costs, including loading, unloading, and transportation costs. Furthermore, we tailored the RRT and termination criteria to the solution approach.

The use of these termination approaches together with different parameter settings for intensification/diversification allows a very flexible approach to either focus on runtime or solution quality. Greater diversification can be achieved in the removal process by removing a large number of orders r and high randomization with large α, in the insert process with a high k value, and in the RRT with a high limit of unsuccessful runs and a high accepted deviation D from the best current solutions.

Table 2. Overview of Numerical Tests

Section	Model	Solution approaches	Data applied
6.1	MCVRP	LNS (Derigs et al. 2011)	Benchmark data
6.2	MCVRP_LU	LNS	Randomly generated data
6.3.1	MCVRP_LU, MCVRP*, SCVRP_LU	LNS	Case study
6.3.2	MCVRP_LU, MCVRP*	LNS	Randomly generated data

6. Numerical Applications

This section presents numerical experiments to evaluate our solution approach and the benefits of the extended model. We compare our modeling approach (MCVRP_LU) with alternative models where loading/unloading costs are disregarded (denoted as MCVRP) or added ex post (denoted as MCVRP*). Furthermore, we analyze the different solution approaches (our LNS, approach proposed by Derigs et al. 2011) and apply these to different data settings (benchmark data, randomly generated data, and case study). Furthermore, to assess the effects of using MCVs instead of SCVs, we compare the MCVRP_LU to a SCVRP_LU, where only one compartment is allowed. Table 2 summarizes the different tests.

In Section 6.1, we apply our solution approach to the benchmark data available in the literature (see Derigs et al. 2011), but without loading/unloading costs. This allows us to demonstrate that our algorithm generates reasonable results and is therefore suitable. In Section 6.2, the performance of our approach is examined for randomly generated problem instances. Here, we study the results of our LNS algorithm for larger problem sizes to obtain further insights into runtime and the influence of different data settings. Finally, Section 6.3 analyzes the effect of introducing loading/unloading costs by comparing the MCVRP_LU results with the MCVRP that only minimizes transportation costs. This has been carried out in two steps. First, we used data obtained from a case study to research the effect of the additional cost factors in practice. Second, we compare the two models for randomly generated data to gain deeper insights into the main drivers for potential savings and derive managerial rules.

Each analysis performed is based on 100 applications of the LNS for the specified problem instance. The computational results were obtained on a 1.8 GHz PC with 8 GB RAM running on Windows 8.1. The implementation was realized in Java. The algorithm-specific parameters are chosen in the following way. For the Shaw Removal, the weights for the calculation of the similarity measure R_{os} were set to $\phi = \omega = 0.4$ and $\psi = 0.2$. This choice of weights is based on the higher influence of distance costs and product segments compared to order size. The value for the selection in the ranking is set to $\alpha = 4$. Good results have been obtained using this setting. The number of removed items r is chosen randomly using a uniform distribution with limits dependent on the respective problem size. The degree for the regret-k operator has been set at $k = 2$ as a standard value. Furthermore, the termination limit is 2,000 and the limit for a solution reset is 500. The maximum deviation D allowed for the RRT equals 0.9% for runtime tests. The impact of these algorithm-specific parameters is justified and further elaborated using sensitivity analyses.

6.1. Performance Analyses Using Benchmark Data

To verify the effectiveness of our implementation and the eligibility of the chosen operators, we applied our approach to instances provided by Derigs et al. (2011). They extensively evaluated different heuristics for the MCVRP and can be seen as a performance benchmark. We compare the solution of our model—without loading and unloading costs—with the best solutions found by Derigs et al. (2011). The available benchmark set concerning food distribution comprises 75 instances with 10, 25, 50, 100, and 200 customers and two or three product types. As indicated in Derigs et al. (2011), all data can be found at http://www.ccdss.org/vrp/. The main characteristics of the instances are summarized in Table 3.

We compared our results to the best overall solutions generated by the heuristic solver suite used by Derigs et al. (2011). To compare the results for identical applications, it is necessary to characterize the data instances. For our comparison, we separate the simulated instances into three different groups to set the focus on the relevant settings.

- Group A: Comprises 27 instances that have a comparable structure to our problem application in terms of order sizes, number of orders per customer, and number of orders ultimately assigned to a vehicle. In practice, usually not more than six customers and 14 orders are on one delivery tour. This is also the case in these data instances.
- Group B: Comprises 38 instances with multiple orders of the same segment of one customer in the same order cycle (e.g., four orders for frozen, two orders for fresh). In retail practice, such multiple orders of one product segment would be summarized in one order. The practical relevance of such instances is more restricted as a result.
- Group C: Comprises 10 instances with very small order sizes compared to the vehicle capacity so that up to 30 orders may be assigned to the vehicle. These data instances do not relate to our problem setting.

Table 3. Instance Characteristics from Derigs et al. (2011)

Characteristic	Values	Comment
Number of customers	10, 25, 50, 100, 200	—
Number of orders	45–796	Depending on capacity and compartments
Number of compartments/segments	2, 3	—
Vehicle capacity	600, 800, 1,000, 3,000, 9,000	—
Total demand	1,000	Demand split over compartments: (0.53, 0.47) and (0.68, 0.24, 0.08)

Figure 7 summarizes the comparison. For related problem settings as in Group A, we were able to show that the average deviation from the best solutions is only 0.14%. Additionally, taking out the two instances with the highest deviation (around 1.0%), the average deviation of the remaining instances equals 0.05%. For Group B, an average deviation of 0.21% was achieved, and Group C shows a deviation of 1.1%. Regarding the whole range of instances tested, for 16 instances a better solution and for another 18 the same solution was found, as provided by Derigs et al. (2011). Contemplating the numerous additional operators used by Derigs et al. (2011), it was thus shown that our leaner solution algorithm within the governing process constructed is able to produce fairly good results.

Assessing the overall structure of the benchmark data, we would like to highlight several points. First, the data is based on identical order sizes and product segments across all customers. Second, the set is limited to 200 customers and up to three product segments, yet retailers may serve 400 or more customers with up to four or five different temperature-specific segments. Third, the data do not include loading/unloading costs, as this was not part of the decision problem.

6.2. Performance Analyses with Randomly Generated Instances

6.2.1. Data Simulation and Parameter Setting. We leveraged data from a case study with a large German retailer to inform the data-generation process. To fully assess the computational efficiency of our approach, we randomly generated new problem instances with the intention of mapping actual requirements given in practice. This involves more customers and product segments, as well as an individual order structure for each customer. This means that all customers neither order the same total volume nor submit orders for the same segments. Moreover, standardized truck capacities for our field of application have been chosen. For all tests, vehicles with a capacity of 33 TUs were used.

Test instances of differing sizes have been generated for each analysis with regard to the number of customer locations and/or orders. Every order comprises one product segment, and every customer places at least one order. The minimum order size is one TU. The *number of customer locations* n, *number of orders* $|O|$, and *number of product segments* $|P|$ have been taken as controllable parameters, and their respective values are denoted in Sections 6.2.2–6.2.4. The *order quantity* was randomly chosen and is indicated in each section using average order size. The actual orders of all product segments at all locations were generated by means of the following procedure. According to the number of total orders and customers, a list of all orders was created where each order had a randomly assigned number within a minimum and maximum order quantity. For example, for an average of seven TUs, orders between one and 14 TUs were chosen. Product segments were randomly allocated to orders following a predefined share of each segment and allowing at most one order of each product for each customer. The *distance* between customer locations is also randomly generated, and the ranges are indicated in each section. The specific parameters for distance, loading, and unloading costs are calculated as follows. The *distance costs* include all relevant costs such as depreciation, insurance, truck driver, maintenance, diesel, etc. Fixed costs were allocated based on lifetime mileage of a truck to obtain costs per distance unit. Time parameters for the *loading and unloading costs* were obtained by empirical data collection. These values originate from the process analysis realized in cooperation with the retailer. In this analysis, representative tours have been accompanied and the times for the different processes have been measured by a time and motion study. We refer to our description in Section 2 for the processes and additional handling involved. All measurements of our case study for loading/unloading costs have been translated into monetary terms by evaluating the times obtained with the corresponding personnel costs. The exact parameters are subject to nondisclosure agreements with the retailer. We therefore report the relative cost parameters in relation to the distance costs $cost_{ij}$ per distance unit. The *loading costs* of one compartment are a multiple of 2.60 of the distance costs of one distance unit (e.g., if a transport of 1 km costs €1, the costs of loading one compartment are €2.60). The *unloading costs* of an SCV are a multiple of 2.05. The further cost ratios are given in Table 4. Note that these values may be retailer- and DC-specific and may only serve as a guideline for the cost parameters in a general setting. For other DCs these may vary significantly—e.g., if the temperature-specific warehouses are not at

Figure 7. (Color online) LNS Algorithm vs. Best Solution of Derigs et al. (2011)

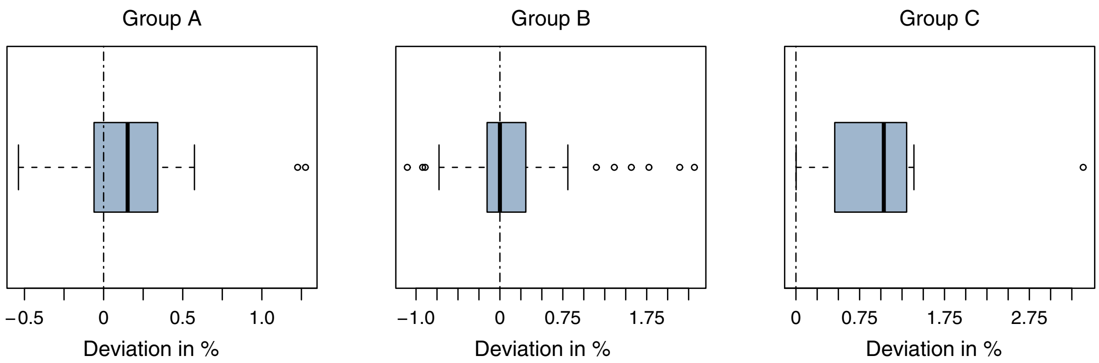


Table 4. Applied Cost Factors for MCVs (in % of SCVs)

Number of compartments c	Loading costs per compartment c	Unloading costs per customer j
2	206	112
3	306	112
4	406	112

Table 5. Average Runtime for Increasing Problem Sizes (in Seconds)

Customer n	10		50		100		200		400	
Order o	20		100		200		400		800	
Segments $\|P\|$	2	4	2	4	2	4	2	4	2	4
$r_{\min}$	2		5		5		5		5	
$r_{\max}$	5		15		30		50		100	
Savings runtime[a]	<1		<1		<1		<1		<1	
LNS runtime	<1	<1	14	4	25	26	144	144	197	285
Total runtime	<1	<1	14	4	26	27	145	145	198	286

[a] Equal for two and four segments/compartments as savings does not consider the number of compartments.

the same location and there is a longer travel time between gates.

All of the instances used and the corresponding solutions for our tests with randomly generated data can be found at http://www1.ku.de/wwf/pw/forschung/.

6.2.2. Runtime Development for Large, Randomly Generated Instances.

Table 5 summarizes the results and further information on the instances for the second set of experiments, where a set of large instances with up to 400 customers was generated. Since the number of removed orders r within the LNS has a great influence on computational times, the runtime is analyzed in relation to r. For all runtime analyses, the distances between any two customers and/or the DC are set at between 11 and 80 km and the average order size has been fixed at seven TUs.

The Savings algorithm performs very well for all instances and therefore provides fast initial solutions as desired. The runtime development of the LNS shows a moderate increase. The first increase can be observed for the set of 200 customers. However, with a runtime of around 2.4 minutes, this is still an acceptable result. It also holds true for the instances with 400 customers. Thus, it can be confirmed that the algorithm provides results for practically relevant problem sizes within reasonable time boundaries.

6.2.3. Sensitivity Analysis with Algorithm-Specific Parameters.

As further studies have shown, the increase in runtime for the LNS is mainly driven by the parameter setting and not by the problem size. Therefore, a sensitivity analysis on the number of removals, the degree of regret, and the solution deviation allowed was performed to examine the corresponding runtime and solution performance. For the parameter analyses, the set of 200 customers, 400 orders, and two product segments introduced for the runtime analysis in Section 6.2.2 was chosen as the standard set. All further instance information is therefore as given above. The results are displayed in the enumeration below and summarized in Table 6.

(1) Shaw Removal: number of removals r. As the main driver for the operations to be performed within the LNS, r has a significant influence on the runtime as well as the solution quality of the LNS. The variation of runtime values grows rapidly for a higher r. Best results can be observed using a random choice of r. This can be attributed to the combination of intensification, using a small number of removals and diversification by using a higher value for r.

(2) Regret-k Insertion: degree of regret k. The average runtime decreases as the degree of regret increases. As the total costs are also shown to be the most excessive for the highest regret value, this can be justified by a decreasing number of iterations performed by the LNS. With a higher value for k, the algorithm ceases to find solution improvements earlier and therefore the termination limit is reached faster. It clearly shows that the degree of regret should not be set too high as this leads to a significant decline in solution quality.

Table 6. Sensitivity Analysis

	Runtime (seconds)			Change in total costs[a]	
Number of removals r	Ø LNS	Min. LNS	Max. LNS	Overall min (%)	Ø Min. (%)
5	6	2	14	+2.5	+3.6
25	94	28	202	—	—
50	371	145	808	−0.1	+0.1
Random	144	66	280	−0.3	+0.2
Degree of regret k					
2	94	28	202	—	—
4	77	19	211	+0.9	+3.4
8	18	11	236	+5.9	+7.5
Allowed deviation D					
0.4%	113	45	200	−4.0	−4.0
0.6%	111	50	238	−2.7	−2.6
0.9%	94	28	202	—	—
1.2%	68	20	127	+1.9	+2.4
1.4%	52	18	99	+3.1	+3.5

[a] Reference values are $r = 25$, $k = 2$, and $D = 0.9\%$.

(3) Record-To-Record Travel: allowed solution deviation D. Runtime decreases with an increasing value for the deviation allowed. The savings in minimal costs is remarkable when comparing the different deviation values. It lies between −4.0% and +3.1% compared to the reference setting of the RRT. Additional tests with a smaller instance of 100 customers have shown that the influence of D is highly sensitive to problem size and cost. More specifically, a higher value for D showed better results for smaller instances.

Summary. In general, the LNS algorithm reveals lower costs for iterations with longer computation times. However, the positive correlation of runtime and minimum costs cannot be regarded as a rule. Only runs with a significantly lower runtime showed inferior results. The remove operator r has a significant impact on runtime and minimum costs. Overall minimum costs could be obtained by a random choice of r. The degree of regret k has a limited impact on runtime but a significant impact on minimum costs. The acceptance level of the RRT has a moderate impact on runtime but no general impact on minimum costs can be observed.

6.2.4. Sensitivity Analysis to Assess Impact of Order Structure. In addition to the parameter analysis, the last runtime test focuses on a change in order structure for the test instance used. Three product segments were used for this, and the average order size has been modified. Except for the order sizes, the same data set as in Section 6.2.3 is applied.

A change in order size has shown a significant impact. As displayed in Table 7, a decrease in order quantity leads to an increase in runtime for the LNS. For an average of nine TUs, the algorithm requires the shortest average runtime at only 1.9 minutes, whereas an average size of three TUs at five minutes consumes more than twice as much computational time. This is due to the greater number of insertion options for lower order quantities. With a lower average size, more orders can be loaded on a single truck and therefore more order permutations are possible. Also, it is more likely that an improved solution can be achieved with a higher number of possible insertions, even though improvements may be small.

6.3. Experiments on the Effects of Loading and Unloading Costs

Sections 6.3.1 and 6.3.2 analyze the effect of our model extension by integrating loading and unloading costs. The first test is based on a case study, while the second is based on randomly generated data to generalize the findings.

6.3.1. Experiments with Instances from Practice. We completed a case study with a large German retailer that provided the data set and used the learning from our models to create tour plans. The data set covers a representative week of orders placed with a DC. Six different instances have been examined, one for each weekday. In total, this involves orders of 406 customers for four different product segments and an order volume of over 4,500 orders. The maximum number of orders per day amounts to over 900 orders by nearly 400 customers. The capacity of all vehicles is

Table 7. Runtime Analysis of Varying Order Sizes on 100 Applications

	Runtime (seconds)		
Ø order size	Ø LNS	Min. LNS	Max. LNS
3	168	72	329
7	41	18	129
9	24	16	61

33 TUs, and the availability of enough trucks to cover the entire customer demand was assumed. The average order size across all product segments and weekdays amounts to around six TUs. Furthermore, distances from the DC to customers range between 1 and 346 km. Cost factors are used according to the description above. The algorithm-specific parameter setting for the LNS equals the setting for the examples with randomly generated data, and the average LNS runtime across all days is 249 seconds.

The use of MCVs yields significantly better solutions than SCVs (see Table 8, Column 2), with a total cost savings of 6.3%. This represents an annual savings due to the use of MCVs versus SCVs of more than €0.5 million for deliveries from this DC. With an average of around six TUs, out of a total of 140 vehicles, only 16 trucks with a single compartment setting are used in the best solution found using MCVs. This shows that it is worth integrating the loading/unloading costs into the decision model. Not integrating them results in 0.9% higher costs (see Table 8, Column 3). Additionally, Table 9 displays the share of each cost factor taking into account the total costs of the example week. As SCVs only approach one shipping gate, it is natural that the share of loading costs is very small compared to MCV deliveries. Beyond that, our approach reduces the share of loading and unloading costs to reach a better overall solution compared to the MCVRP*.

6.3.2. Intensified Experiments with Randomly Generated Instances. Following the implications of our case study, we embark on examining and generalizing the impact of loading/unloading costs. We use different data settings to analyze the magnitude and derive managerial insights based on these configurations. To test the influence of varying loading/unloading cost factors, we used the data obtained from the case study (this is denoted as "standard") and a rise in these costs by 50% and 100%. As the test instances were generated with respect to orders per customer, the number of customers does not add up to exactly 200 or 100. Also, as we generated the data with respect to the share of segments and order sizes comparable to the case study, Tests C.1 and C.2 are not rounded to hundreds. We apply settings with more than two compartments to highlight the differences. The settings of the tests performed are summarized in Table 10. The results of MCVRP_LU versus MCVRP* are compared in Table 11, where negative entries represent lower values for MCVRP_LU in comparison to the MCVRP*.

Table 8. Results of the Case Study

	MCVRP_LU vs. SCVRP_LU	MCVRP_LU vs. MCVRP*	MCVRP_LU
Weekday	Change in total costs of SCVRP_LU (%)	Change in total costs of MCVRP* (%)	Ø number of compartments
1	−5.5	−1.0	2.17
2	−13.2	−0.9	2.36
3	−7.2	−0.8	2.32
4	−4.2	−1.5	2.17
5	−3.4	−0.6	2.03
6	−4.6	−0.4	1.73
Total	−6.3	−0.9	2.15

Table 9. Share of Loading, Unloading, and Transportation Costs Across the Complete Example Week

	SCVRP_LU			MCVRP*			MCVRP_LU		
Cost details	lc	tc	ulc	lc	tc	ulc	lc	tc	ulc
Total week (%)	1.6	92.4	6.0	4.0	90.3	5.7	3.6	91.1	5.3

Test A.1: Reference case. In the starting problem set, three product segments are used, meaning that customers can place up to three different orders. The direct integration of loading/unloading costs results in up to 1.1% lower total costs and up to 4.0% longer travel distances for MCVRP_LU versus MCVRP*. Note that the transportation costs for the MCVRP* stay constant. The cost share of loading/unloading for the MCVRP* amounts to 10% of the total costs for each process. For the MCVRP_LU this share is reduced to loading costs of 7% and unloading costs of 8% of total costs. A more detailed discussion of this effect is presented below.

Test A.2: Effect of varying order sizes. Next, a change in order quantity for the same customer setting as in Test A.1 is analyzed. For the tests of altering order quantities, order sizes with an average quantity of around nine, five, and three TUs instead of seven TUs are considered (see Table 12).

There are two effects on the savings potential of MCVRP_LU versus MCVRP* concerning the variation of order sizes and the corresponding loading/unloading costs. First, a decrease in order sizes leads to an increase in savings potential for the standard costs. Second, rising loading/unloading costs lead to inverted results so that the savings potential decreases for lower order sizes but increases for higher volumes. This can be accounted for as follows. On one hand, lower order quantities lead to the use of more compartments as more customers are on one truck. Since the orders of more customers are on a truck, it is more likely that different segments will be required. If, for instance, the average order quantity is assumed to be three TUs together with standard cost factors, the best overall solution found by the MCVRP_LU is made up of a fleet consisting of around 95% of vehicles with three compartments. Having a fleet with virtually only MCVs with the maximum number of compartments, loading costs for MCVRP_LU and MCVRP* are almost equally high. In this case, the savings can only be realized for unloading. With increasing loading/unloading costs,

Table 10. Overview of Test Settings

Test	Orders $\|O\|$	Orders per customer	Customers n	Segments $\|P\|$	Ø order size q	Distances d_{ij}
A.1	400	1–3	199	3	7	11–80
A.2	400	1–3	199	3	3, 5, 7, and 9	11–80
B	400	1–3	199	3	7	20–45[a] and 9–40[b]
C.1	198	1–3	106	3	7	20–45[a] and 9–40[b]
C.2	106	1	106	3	7	20–45[a] and 9–40[b]

[a] Distances from the DC to customers.

[b] Distances between customers.

Table 11. Summary of All Tests for MCVRP_LU vs. MCVRP*, 100 Applications for Each Test

	Min. total costs (%)			Min. loading/unloading costs (%)			Min. travel distance (%)		
Test	Standard	+50%	+100%	Standard	+50%	+100%	Standard	+50%	+100%
A.1	−0.4	−0.4	−1.1	−8/−3	−14/−5	−29/−13	+0.2	+1.4	+4.0
A.2[a]	−0.5	−0.6	−1.0	−5/−4	−11/−7	−21/−13	—	—	—
B	−0.5	−0.8	−2.3	−9/−3	−22/−8	−49/−22	+0.4	+2.7	+8.3
C.1	−0.3	−0.6	−1.8	−7/−2	−16/−6	−37/−16	+0.3	+1.8	+6.1
C.2	−0.6	−5.3	−10.3	−22/−2	−63/−32	−64/−44	+1.1	+3.3	+4.5

[a] Average of different order sizes for each model (see Table 12 for details).

Table 12. Test A.2: MCVRP_LU vs. MCVRP* for Varying Order Settings, 100 Applications

Ø order	Min. total costs (%)			Min. loading/unloading costs (%)		
quantity (in TUs)	Standard	+50%	+100%	Standard	+50%	+100%
9	−0.3	−0.8	−2.1	−8.3/−2.7	−22.3/−9.6	−41.9/−22.5
7	−0.4	−0.4	−1.1	−7.5/−2.6	−14.3/−4.9	−28.7/−13.3
5	−0.5	−0.6	−0.7	−3.0/−4.8	−5.0/−6.0	−11.1/−8.3
3	−0.6	−0.5	−0.3	−0.3/−6.3	−0.3/−7.2	−1.1/−7.6

Table 13. Test A.2: MCVRP_LU vs. MCVRP* by Cost Type, Example with an Average Order Size of 3 TUs

Cost type	Standard (%)	+50% (%)	+100% (%)
Loading costs	−0.3	−0.3	−1.1
Transportation costs	−0.1	+0.4	+1.2
Unloading costs	−6.3	−7.2	−7.6
Total costs	−0.6	−0.5	−0.3

the MCVRP_LU tours will have an even larger share of MCVs with the maximum number of compartments to realize savings with fewer stops.

However, this leads to higher transportation costs compared to the MCVRP*, which cannot be completely compensated by the saving for unloading. As a consequence, the MCVRP_LU yields fewer cost reductions in direct comparison to the MCVRP* for small order sizes and higher loading/unloading costs (see Table 13). On the other hand, loading cost savings can be realized for larger order sizes. Paying respect to higher costs, the extended model provides tours with fewer compartments and a more elaborate compartment mix. Therefore, the cost savings between MCVRP_LU and MCVRP* increase as order quantities grow and loading/unloading costs rise.

Test* B: *Effect of shorter travel distances. Test B applies shorter traveling distances (e.g., for urban deliveries). Total cost savings of up to 2.3% are possible, even with up to 8.3% longer travel distances. As mentioned previously, the proportion of loading/unloading costs is the main driver for differing results (see Table 14).

To conclude Test B, the solution structures of MCVRP_LU and MCVRP* are considered in detail. As Table 15 shows, the structure for the tours changes noticeably for the MCVRP_LU as costs rise. Higher loading/unloading costs imply the use of fewer vehicles with multiple compartments; thus, the number of stops increases, whereas the average number of compartments used declines. This change in tour structure within the MCVRP_LU points to the active alignment to modified conditions offered by the extended model.

Test* C.1: *Effect of shorter travel distances for smaller instances. Similar to Test B, another test with an equal setting but fewer customers was performed. Test C.1 confirms the findings above and cost savings also for smaller problem instances. As fewer customers and orders are considered, the possible cost savings using

Table 14. Test B: MCVRP_LU vs. MCVRP* (in %)

	MCVRP_LU vs. MCVRP*, costs for				Shares of MCVRP_LU costs for		
	Loading	Transportation	Unloading	Total	Loading	Transportation	Unloading
Standard	−9.1	+0.4	−3.0	−0.5	6.1	87.6	6.3
+50%	−22.0	+2.5	−7.5	−0.8	7.4	84.2	8.4
+100%	−48.5	+7.7	−21.7	−2.3	6.2	84.7	9.1

Table 15. Test B: Analysis of Change in Solution Structure for MCVRP_LU

	MCVRP_LU			
Tour information	Standard	+50%	+100%	MCVRP*
Total tours	91	91	91	91
Total stops	256	276	327	268
Ø stops	2.8	3.0	3.6	2.9
Ø compartments	2.5	2.2	1.6	2.7

the MCVRP_LU also decrease. In this case, a maximum total cost decrease of 1.8% can be achieved.

Test C.2: Effect of modified order structure. A final test for a modified order structure has been analyzed. It was assumed that each of the 106 customers only submits one order, leading to a total of 106 orders. This involved each customer choosing between three different product segments, as in the previous test. The variation in order structure leads to a significant increase in potential cost savings. The savings rise significantly especially for larger cost parameters. This progress can be explained by the segment mix of the MCVRP*. As only the distance is considered, there are many trucks using different compartments for a single order of another segment. The MCVRP_LU takes this condition into account. With regard to the solution structure for both models, the best solution for the MCVRP* and highest cost setting only involves three vehicles with a single compartment out of a total of 23 tours, whereas the MCVRP_LU uses a completely different structure with 20 vehicles with a single-compartment setting.

Summary. The tests performed have shown that consideration of the cost factors introduced yields significant saving potentials in many applications. Depending on the average order sizes, distances, and the order structure, a significant total cost saving of between 0.3% and 10.3% can be achieved. The cost factors applied have a very significant influence on the results. To summarize all findings in line with the model comparison, an overview of the savings potential via the use of the MCVRP_LU is displayed in Table 16.

7. Conclusion and Further Areas of Research

The objective of this paper was to develop a model and solution approach to analyze the effectiveness of MCV food distribution. We analyzed the basis for VRP with multiple compartments and decision-relevant costs. An enhanced model, the MCVRP_LU, was introduced to account for loading costs that increase with each additional compartment used, and unloading costs that decrease when product segments of a customer are delivered jointly. For solving the MCVRP_LU, an LNS with well-selected operators and control mechanisms was chosen that provided good solutions. The MCVRP_LU has been tested in terms of computational times and shown to be efficient with regard to its runtime. The average time needed for the LNS is less than five minutes for problem sizes relevant in practice—i.e., 400 customers and four compartments. Furthermore, the number of removals and average order sizes have been identified as the main drivers for runtime.

Integrating loading/unloading costs in the decision model yields significant savings potential. This was first shown by a case study and additionally with experiments using randomly generated data. Subject to the number of compartments, customer distances, applied cost factors, and the order structure, the MCVRP_LU shows a savings potential of between 0.3% and 10.3% compared to the MCVRP*. Beyond this, a comparison with the classic CVRP but including loading/unloading costs stresses the superiority of a distribution system involving MCVs. Here, a cost saving of 6.3% could be achieved for the data considered in line with the case study.

To summarize, the MCVRP_LU model developed has shown significant savings potential in various problem settings. A more accurate evaluation of the VRP is achieved with the introduction of divergent costs for MCV applications. Furthermore, the application made it clear that the use of MCVs is highly advisable, regardless of the given setting.

Future areas of research. There are numerous opportunities for further research in the field of MCVs and the use of the model presented. First, heuristics can always be further developed and tested in different settings. The development of additional solution approaches might be a starting point for further computational research, for the construction heuristics as well as the improvement heuristics. Several extensions of the MCVRP_LU may be considered to enhance compatibility with practical requirements. For this purpose, extensions to account for stochastic demand,

Table 16. Findings for the Development of Saving Potential

Characteristic	Savings potential of MCVRP_LU vs. MCVRP* increases with	Correlation with increasing loading and unloading costs	Corresponding test
Product segments	More product segments	Positive	A.1
Ø order size	Lower average order size	Positive for order sizes ≥ 5; negative for order sizes < 5	A.2
Distance	Shorter distances	Positive	B and C.1
Number of orders per customer	More heterogeneous order structures across customers	Positive	C.2

time windows, a heterogeneous distribution fleet, and multiple periods are possible next steps. Additionally, the MCVRP_LU can be extended to carry out a vehicle selection, choosing between multi- and single-compartment vehicles. This can also be used to determine the optimal fleet when taking vehicle purchasing costs into account. Furthermore, we assume enough capacity for all orders. Retailers may be faced with warehouse, transportation or in-store capacity shortages, where not all orders can be delivered. Introducing nondelivery costs into the objective function could model these kinds of applications. We have only posited tours where each order is entirely assigned to one compartment. In some applications, splitting an order and distributing it among different vehicles may be feasible. Finally, the model could be extended to analyze the impact of multi-temperature deliveries on in-store logistics and inventory, as well as warehouse operations (e.g., because of a higher delivery frequency), due to more flexible delivery patterns (Holzapfel et al. 2016).

Acknowledgments

The authors thank the guest editor and the anonymous reviewers for valuable recommendations that significantly improved the paper.

References

Agrawal N, Smith SA, eds. (2015) *Retail Supply Chain Management: Quantitative Models and Empirical Studies*, Internat. Series Oper. Res. Management Sci. (Springer, New York).

Akkerman R, Farahani P, Grunow M (2010) Quality, safety and sustainability in food distribution: A review of quantitative operations management approaches and challenges. *OR Spectrum* 32(4):863–904.

Amorim P, Meyr H, Almeder C, Almada-Lobo B (2013) Managing perishability in production-distribution planning: A discussion and review. *Flexible Services Manufacturing J.* 25(3):389–413.

Avella P, Boccia M, Sforza A (2004) Solving a fuel delivery problem by heuristic and exact approaches. *Eur. J. Oper. Res.* 152(1):170–179.

Brown GG, Graves GW (1981) Real-time dispatch of petroleum tank trucks. *Management Sci.* 27(1):19–32.

Chajakis ED, Guignard M (2003) Scheduling deliveries in vehicles with multiple compartments. *J. Global Optim.* 26(1):43–78.

Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12(4):568–581.

Derigs U, Gottlieb J, Kalkoff J, Piesche M, Rothlauf F, Vogel U (2011) Vehicle routing with compartments: Applications, modelling and heuristics. *OR Spectrum* 33(4):885–914.

Dueck G (1993) New optimization heuristics. *J. Comput. Phys.* 104(1): 86–92.

El Fallahi A, Prins C, Wolfler Calvo R (2008) A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Comput. Oper. Res.* 35(5):1725–1741.

Fernie J, Sparks L, eds. (2009) *Logistics and Retail Management: Emerging Issues and New Challenges in the Retail Supply Chain*, 3rd ed. (Kogan Page Ltd., London).

Fisher ML, Jaikumar R (1981) A generalized assignment heuristic for vehicle routing. *Networks* 11(2):109–124.

Glatzel C, Großpietsch J, Hübner A (2012) Higher margins through efficient supply chains. *Akzente* (2):16–21.

Golden BL, Raghavan S, Wasil EA (2008) *The Vehicle Routing Problem: Latest Advances and New Challenges*, Oper. Res./Comput. Sci. Interfaces Series (Springer Science+Business Media, New York).

Grünrock-Kern U (2013) Deutscher Logistik-Preis 2013 für Lekkerland. http://www.bvl.de/presse/meldungen/archiv-2013/deutscher-logistik-preis-2013-fuer-lekkerland.

Henke T, Speranza MG, Wäscher G (2015) The multi-compartment vehicle routing problem with flexible compartment sizes. *Eur. J. Oper. Res.* 246(3):730–743.

Holzapfel A, Hübner A, Kuhn H, Sternbeck M (2016) Delivery pattern and transportation planning in grocery retailing. *Eur. J. Oper. Res.* 252:54–68.

Hübner A, Kuhn H, Sternbeck MG (2013a) Retail operations: Why and how retailers benefit from an integrative supply chain management perspective. Wimmer T, Hucke S, eds. *Inspiration, Ideas, Innovation* (DVV Media Group, Hamburg, Germany), 359–439.

Hübner AH, Kuhn H, Sternbeck MG (2013b) Demand and supply chain planning in grocery retail: An operations planning framework. *Internat. J. Retail Distribution Management* 41(7):512–530.

Kuhn H, Sternbeck MG (2013) Integrative retail logistics: An exploratory study. *Oper. Management Res.* 6(1–2):2–18.

Lahyani R, Coelho LC, Khemakhem M, Laporte G, Semet F (2015) A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. *Omega* 51:1–10.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Sci.* 43(4):408–416.

Melechovský J (2013) A variable neighborhood search for the selective multi-compartment vehicle routing problem with time windows. Yeomans JS, Montemanni R, Nordlander TE, eds. *Proc. 5th Internat. Conf. Appl. Oper. Res.*, Lecture Notes Management Sci., Vol. 5 (Tadbir, Vancouver), 159–166.

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2010) A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Comput. Oper. Res.* 37(11): 1886–1898.

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2011) Constructive heuristics for the multi-compartment vehicle routing problem with stochastic demands. *Transportation Sci.* 45(3):346–363.

Muyldermans L, Pang G (2010) On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *Eur. J. Oper. Res.* 206(1):93–103.

Pirkwieser S, Raidl GR, Gottlieb J (2012) Improved packing and routing of vehicles with compartments. Moreno-Díaz R, Pichler F, Quesada-Arencibia A, eds. *Comput. Aided Systems Theory—EUROCAST* 2011, Lecture Notes Comput. Sci., Vol. 6927 (Springer, Berlin Heidelberg), 392–399.

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(2):297–330.

Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Sci.* 40(4):455–472.

Shaw P (1997) A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, Department of Computer Science, University of Strathclyde, Glasgow, Scotland.

Toth P, Vigo D (2014) *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series Optim., 2nd ed. (SIAM, Philadelphia).