Product-oriented time window assignment for a multi-compartment vehicle routing problem

Sara Martins[a,*], Manuel Ostermeier[b], Pedro Amorim[a], Alexander Hübner[b], Bernardo Almada-Lobo[a]

[a] *INESC TEC and Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, s/n, Porto 4600-001, Portugal*
[b] *Supply and Value Chain Management, Technical University of Munich, Schulgasse 22, Straubing 94315, Germany*

Besides fuel and waste distribution, one core application of multi-compartment vehicles (MCVs) is the distribution of groceries, as they enable retailers to jointly transport products with different temperature requirements, thus reducing the number of visits to a store. Grocery stores usually define preferable time windows that depend on the temperature of products (for example, fresh products in the morning) to indicate when deliveries should occur to better plan their in-store operations. Distribution planning therefore needs to take these preferences into consideration to obtain consistent delivery times. This work extends the research on multi-compartment vehicle routing problems (MCVRPs) by tackling a multi-period setting with a product-oriented time window assignment. In this problem, a fleet of MCVs is used for distribution and a unique time window for the delivery of each product segment to each store is defined consistently throughout the planning horizon. An ALNS is proposed to solve the product-oriented time window assignment for MCVRP. Daily and weekly operators are developed respectively focusing on the improvement of routing aspects of the problem on each day and aligning the time window assignment consistently throughout the planning horizon. The approach is tested on benchmark instances from the literature to demonstrate its effectiveness. We also use direct information from retail practice and enhance this with simulated data to further generalize our findings. The numerical experiments demonstrate that planning consistent MCV distribution leads to better overall solutions than the ex-post time window assignment of daily plans, facilitating more on-time deliveries.

1. Introduction

This paper introduces a multi-compartment vehicle routing problem (MCVRP) for the assignment of product-oriented time windows. The considered problem formulation with multi-compartment vehicles (MCVs) can be found in grocery distribution (e.g., supply of stores with different temperature zones), waste collection (e.g., glass waste) and fuel distribution (e.g., supply of petrol stations) where repetitive delivery cycles are applied and customers rely on consistent time windows. Our problem is motivated by an application in grocery distribution where a high product variety with particular temperature requirements needs to be managed and an efficient supply chain is essential (Klingler, Hübner, & Kempcke, 2016). The corresponding routing problem states a tactical decision problem for retailers as it defines master routes for the daily routing.

Products with similar characteristics and temperature requirements are usually denoted as product segments or simply segments in grocery distribution. In the past, only one product segment could be transported within the same vehicle as the temperature could only be set up at one level at a time. However, MCVs have been developed for grocery transport in recent years. These are able to split the loading area flexibly into compartments with different temperatures whilst there is no loss in capacity as the compartments are continuous in size. However, the joint transportation of segments forces retailers to decide which segments are (or are not) supplied jointly for each store. This explicitly implies a higher variety of possible delivery times for the stores. For example, ambient products can be delivered jointly with the fresh products in the morning, instead of a separate delivery of ambient products later during the day. This impacts store operations that need to align their operations to the delivery schedule to make sure their resources are available for unloading, replenishment and

[*] Corresponding author.
E-mail addresses: sara.martins@fe.up.pt (S. Martins), manuel.ostermeier@tum.de (M. Ostermeier), pamorim@fe.up.pt (P. Amorim), alexander.huebner@tum.de (A. Hübner), almada.lobo@fe.up.pt (B. Almada-Lobo).

stocking, which impacts staff scheduling, backroom capacity management and on-shelf availability. This is especially significant as store resources are scarce and shared by different activities. Additionally, stores need to manage the available inventory as shelf inventory which is crucial for sales (see Hübner & Schaal, 2017). The definition of time windows helps to manage the on-hand inventories at stores as the time until the next supply is known (Holzapfel, Hübner, Kuhn, & Sternbeck, 2016). The scheduling of deliveries therefore needs to be defined according to the stores time window requirements (independent from joint or separate deliveries across products). Delivery time windows have to be predefined and fixed for a given period to efficiently coordinate store resources during the day. This requires consistent delivery time windows for each product segment throughout the planning horizon, ensuring a smooth supply of the stores.

While current VRP literature usually assigns time windows to customers (Spliet & Desaulniers, 2015; Spliet & Gabor, 2014), a more specific product oriented assignment is required when dealing with multiple products with distinct characteristics such as grocery products. It is not sufficient to consider multiple delivery time windows per customer (Belhaiza, Hansen, & Laporte, 2014). Instead it is necessary to define which of the available time windows each product will be assigned to, ensuring consistent deliveries for a given planning horizon. In combination with the transportation in MCVs, this raises the question of when each product should be supplied, and whether the same time window should be assigned to different products to enable joint delivery. Our work addresses this special variant of periodic VRPs with MCVs and a product-oriented time window assignment, which is consistent throughout the planning horizon. To the best of our knowledge, there is not yet an MCVRP that integrates the consistent time window assignment in a multi-period setting. Consequently, our main contributions are as follows. Firstly, a mixed integer programming model defining the product-oriented time window assignment for MCVRP (PTWA-MCVRP) is proposed. The defined model enables a tactical planning of master routes for the distribution in retailing. Secondly, an adaptive large neighborhood search (ALNS) is designed to solve the problems characteristics with specialized, innovative operators. Its effectiveness is tested on both benchmark instances and simulated data informed by retail practice. Thirdly, the effects of consistent deliveries and product-oriented time windows with MCVRPs are analyzed.

The remainder of this paper is organized as follows. Section 2 provides a detailed description of the problem and its related literature. Section 3 describes the formulation of the mathematical model. The ALNS algorithm developed is explained in Section 4. Numerical experiments are carried out in Section 5. Finally, our findings are summarized in Section 6.

2. Distribution process, requirements and related literature

In this section we first describe the overall distribution planning process for retailers before highlighting the implications of using MCVs for transportation. Further, we analyze why grocery retailers need to consider multiple periods and consistent deliveries across product segments. This information has been collected in joint projects with European retailers. This builds the foundation of the literature review that follows and accounts for how the areas of further research have been derived. The terms customers and stores are used as equivalents in our context.

2.1. Description of the planning problem

Distribution of groceries. Grocery retailers need to simultaneously manage four to five different temperatures (e.g., frozen, chilled and ambient) across their logistics subsystems. The majority of products are distributed via an DC (Hübner, Kuhn, & Sternbeck, 2013; Martins, Amorim, & Almada-Lobo, 2017). The distinct products can be allocated to the same DC, but separate warehouse zones at different temperatures are required to prevent spoilage. The same reasoning applies to the transportation process, during which the preferred temperature for each product needs to be maintained to guarantee high product quality and to adhere to legal regulations.

Distribution with MCVs. Using MCVs poses some new challenges for the planning as the joint distribution of segments influences not only the transportation process but also the upstream and downstream supply chain operations (Hübner & Ostermeier, 2018). On the one hand, different gates have to be approached by an MCV to collect different segments from distinct DC temperature zones. This leads to an increase in loading costs that depend on the number of segments assigned to a tour and therefore on the number of compartments needed on each vehicle. On the other hand, separate deliveries with a single-compartment vehicle for each segment may be avoided, reducing the number of visits to a store and total travel distance.

Application of product-oriented time windows for store operations. Store replenishment usually takes place at certain time windows during the day. These time windows depend for example on the general replenishment policy of certain segments (e.g., fresh products need to be replenished before or on opening of the store), availability of replenishers (e.g., external replenishers or part-time workers may only be available at certain times or store employees during low store traffic times) and further store requirements and opportunities (e.g., the possibility of using the backroom for intermediate storage). To enable smoother operations, plan capacity for store operations and ensure appropriate store inventory levels, grocery retailers rely on consistent product-oriented time windows. These time windows ensure that each product segment will always arrive during the same time window along the planning horizon (i.e., the full week). In practice, retailers usually determine master routes based on the demand of an average week as they require consistent solutions for such routes regarding average weeks. These routes can then be used as template for the daily planning with only minor adjustments. Master routes are usually determined annually or each second year. Furthermore, the definition of delivery days for each store (i.e., delivery patterns) mainly depends on costs for store operations and only partially on costs for picking and transportation. In decision models to define delivery patterns (see e.g., Holzapfel et al., 2016; Taube & Minner, 2018), the tour costs are usually approximated without defining the actual tours. Yet, in retail practice, the delivery patterns can also be defined in a precedent step of the actual routing (Kuhn & Sternbeck, 2013). This means that the stores have defined days to order and receive certain segments.

An example of the assignment of product-oriented time windows for different store-segment pairs is depicted in Fig. 1. It illustrates two examples of routing schedules, where time window assignments are considered with and without consistent deliveries. Accordingly, segment delivery patterns are provided on the left-hand side of Fig. 1. For example, segment A (dark square) is delivered every day, while segment D (dotted square) is only delivered on Mondays, Wednesdays and Fridays. Two possible solutions for the time window assignment are presented. The schedule on the left represents a solution where consistency is not taken into account, while the one on the right contains a consistent schedule. In the first case, segments are delivered at different times of the day over the week. In the second case, deliveries within the same time bounds are guaranteed across the entire planning horizon. It

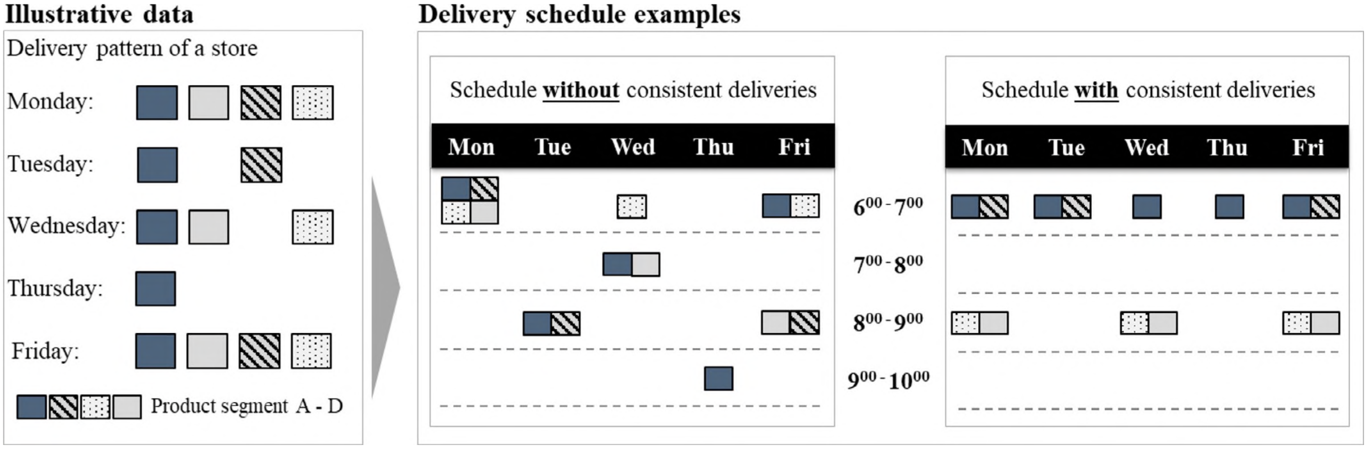

Fig. 1. Illustrative example of routing schedules with and without consistent deliveries.

shows that segments A & B are supplied between 6 and 7 am every day of the planning horizon. Likewise, deliveries for segments C & D always take place between 8 and 9 am. Clearly, this schedule enables stores to plan their resources according to the consistent plan, allowing efficient processing of deliveries.

Consequently, the aim of the PTWA-MCVRP is to determine an individual time window for each segment that a store has ordered and to use this time window consistently for the complete planning horizon. Each segment is thereby considered independently of all other segments for the time window assignment. However, the same time window can be selected for different segments as stores usually order more than one segment a day. This also means that the joint delivery of multiple segments is possible if they are requested on the same day. Yet, the requirement for consistent delivery times applies to all segments. Ultimately, the product-oriented time window is motivated to reduce the planning complexity for stores and increase the stores satisfaction with its supplier as a result. Hereafter, we refer to a given segment of a store for which a product-oriented time window has to be determined as store-segment pair.

In retail practice, the assignment of product-oriented time windows underlies some further requirements and rules. Time window restrictions are not strict and both early and late deliveries to the stores are possible but undesirable, having a negative impact on store operations. Usually, the delivered products are dispatched to the sales area for replenishment and the excess is stored in the backroom area for later replenishment. The backrooms are designed to store just a portion of the deliveries, ensuring future replenishment, and early deliveries can trigger storage problems at this stage if the deliveries are not processed immediately upon arrival, specially refrigerated products (Pires, Pratas, Liz, & Amorim, 2017). Late deliveries, however, lead to idle times of the personnel assigned to the receiving activity and might delay the replenishment of shelves with the risk of causing stockouts. We note that in-store operations yield the highest share of operational costs within the internal supply chain of a retailer, accounting for up to 50% (see e.g., van Zelst, van Donselaar, van Woensel, Broekmeulen, and Fransoo (2009) and Kuhn and Sternbeck (2013)).

2.2. Related literature

The PTWA-MCVRP deals with two main groups of decisions: (i) routing decisions and (ii) consistent time window assignment decisions. These decisions relate to three streams of VRP variants that will be discussed below. Since the routing decisions are defined taking MCVs into account, the PTWA-MCVRP clearly extends literature on MCVRPs. The time window assignment decisions derive from an extension of the time window assignment vehicle routing problem (TWAVRP) and the consistent vehicle routing problem (ConVRP).

Multi-compartment vehicle routing problems. MCVRP literature focuses mainly on its applications to fuel distribution (e.g., Avella, Boccia, & Sforza, 2004; Coelho & Laporte, 2015; Cornillier, Boctor, Laporte, & Renaud, 2008), waste collection (e.g., Henke, Speranza, & Wäscher, 2015; Muyldermans & Pang, 2010; Reed, Yiannakou, & Evering, 2014) and food distribution (e.g., Chajakis & Guignard, 2003; Hübner & Ostermeier, 2018). Most of the works on MCVRP assume that customers can only be served by one vehicle (e.g., Abdulkader, Gajpal, & ElMekkawy, 2015; Chajakis & Guignard, 2003; Reed et al., 2014) and/or that the number of compartments and their size are fixed (e.g., El Fallahi, Prins, & Calvo, 2008; Muyldermans & Pang, 2010), which are too restrictive for our setting.

Derigs et al. (2011) are the first to consider flexible compartment sizes with multiple deliveries to customers. This new feature creates a more general MCVRP by adding decisions on the number and size of compartments to the problem. Henke et al. (2015) tackle the MCVRP with discrete flexible compartments, instead of continuous ones as in Derigs et al. (2011). The authors allow the number of compartments to be smaller than the number of products to be collected and apply a variable neighborhood search to the problem. Later, Koch, Henke, and Wäscher (2016) propose a genetic algorithm for the same problem and Henke, Speranza, and Wäscher (2017) develop an exact method for the problem (a branch-and-cut algorithm). Hübner and Ostermeier (2018) study the distribution of groceries with flexible MCVs, incorporating the operational costs of the loading and unloading processes. The authors propose a large neighborhood search (LNS) to solve the problem. Ostermeier and Hübner (2018) present a vehicle selection model for the MCVRPs that analyzes under which conditions single- or multi-compartment vehicles are more efficient.

The MCVRP was also extended by some authors to incorporate time window restrictions that consider fixed compartment sizes. Kaabi and Jabeur (2015) describe an MCVRP where customer orders are composed of different products with associated profits, which are collected once the customer is visited within their time window. The authors propose a hybrid approach combining a genetic algorithm and an iterated local search to solve the problem. Kabcome and Mouktonglang (2015) consider time windows as soft constraints but set an upper bound for the violation of time windows as hard constraint. The authors use a commercial software to solve small instances exactly. Although both works consider time window restrictions, they are incorporated as input parameters rather than decision variables. A model and solution approach that integrates the MCVRP with time window assignment decisions has not yet been analyzed in the literature, but has practical relevance as described in above.

Time window assignment & consistent vehicle routing problems. Spliet and Gabor (2014) introduced the first variant of the TWAVRP. They define a problem where the assignment of time windows to each customer is performed before the actual demand is known. Afterwards, when the order is submitted, a vehicle routing schedule is made to comply with the assigned time windows. They assign time windows of prespecified width from a set of endogenously known time windows. The authors develop a branch-price-and-cut algorithm to find the optimal expected traveling time. In a subsequent extension, Spliet and Desaulniers (2015) consider the assignment of time windows from a predefined set of time windows. They propose a branch-price-and-cut algorithm and different column generation heuristics. Spliet, Dabia, and van Woensel (2017) introduced time-dependent travel times to the TWAVRP, focusing on predictable variations. The authors develop a branch-price-and-cut algorithm, using an exact labeling and a tabu search heuristic to solve the pricing problem. Jabali, Leus, Van Woensel, and De Kok (2015) consider a similar problem, in which travel times are stochastic but the demand is deterministic. The goal is to define a single route plan that should be fixed for all days of the year together with the time windows assignment. The authors propose a tabu search algorithm to minimize the total traveling costs and expected earliness and tardiness penalty costs, assuming soft time windows. Feillet, Garaix, Lehuédé, Péton, and Quadri (2014) present a bi-objective time-consistent VRP aimed at minimizing - besides the total travel time - the maximum number of time classes in which a customer is visited. The authors solve the problem with an LNS. Subramanyam and Gounaris (2016) develop an exact solution for the consistent traveling salesman problem (ConTSP) aimed at identifying the minimum-cost set of routes that a single vehicle should follow for a given period, ensuring that the customers are visited at roughly the same time of the day. Subramanyam and Gounaris (2017) extend the previous work from an TSP to an VRP variant and show that each scenario of the TWAVRP stochastic model can be reduced to an VRP variant known as a consistent vehicle routing problem (ConVRP). The authors adapted an exact algorithm for the ConVRP and solved the TWAVRP benchmark instances.

The ConVRP is a multi-period problem proposed by Groër, Golden, and Wasil (2009) that aims to design consistent routes over a given planning horizon. Kovacs, Golden, Hartl, and Parragh (2014a) describe three different types of consistent routing: (i) arrival-time consistency, which ensures visits to customers at roughly the same time of the day, (ii) person-oriented consistency, which means that customers are visited by the same driver, and (iii) delivery consistency, where customers receive roughly the same quantity of goods. The arrival-time consistency requirement of the ConVRP is similar to an TWAVRP, as both problems define an interval of time within which customers visits should occur. Most of the literature on ConVRP focuses on arrival-time and driver consistency. The first approaches solving the problem are based on template concepts, wherein template routes are built considering the frequent customers, and afterwards a daily plan is derived to include the remaining customers to be visited on each day (Groër et al., 2009; Kovacs, Parragh, & Hartl, 2014b; Tarantilis, Stavropoulou, & Repoussis, 2012). Groër et al. (2009) propose a multi-start solution construction combined with a Record-to-Record travel local search metaheuristic algorithm, and Tarantilis et al. (2012) propose a tabu search algorithm. Kovacs et al. (2014b) present an ALNS to solve the problem and analyze the variant of allowing later departures from the depot. Sungur, Ren, Ordóñez, Dessouky, and Zhong (2010) also use the template concept for the courier delivery problem with uncertainty, where the problem is solved by a master and daily scheduler heuristic (MADS).

Kovacs, Golden, Hartl, and Parragh (2015a) generalize the ConVRP, allowing the customer to be visited by a limited number of drivers and penalizing the arrival-time variation in the objective function. The authors propose an LNS algorithm to solve the problem. They found that both driver and arrival-time consistency have a small impact on the fleet size. Kovacs, Parragh, and Hartl (2015b) and Lian, Milburn, and Rardin (2016) study the multi-objective ConVRP, where driver and arrival-time consistency are considered as objectives, besides the traveling cost. Both works analyze the trade-off between traveling costs and service consistency, and develop a multi-directional local search (MDLS), combined with an LNS to approximate a Pareto frontier. Kovacs et al. (2015b) also propose two exact approaches based on the ϵ-constraint framework and state that, on average, it is possible to achieve 70% better arrival-time consistency by increasing the traveling costs by not more than 4%.

We will leverage the current literature in different ways. First of all, we will propose an ALNS that uses daily operators based on Derigs et al. (2011) and Hübner and Ostermeier (2018) approaches and incorporates the operational features of the latter. Secondly, we will use the concept of consistency regarding the time window assignment. A maximum width between arrivals will be defined as for the ConVRP, but not all arrival times are preferable. This means that we need to define a set of time windows from where the assignment is made in order to incorporate the stores preferred delivery times.

3. Problem definition and model formulation

The PTWA-MCVRP is defined on a complete undirected, weighted graph $G = (N, E)$, where $N = \{0, 1, \ldots, n\}$ is the set of nodes and $E = \{(i, j) : i, j \in N\}$ is the set of edges. Node $i = 0$ represents the DC location. Each edge $(i, j) \in E$ is associated with a traveling cost tc_{ij} and a traveling time tt_{ij}. It is assumed that all traveling costs satisfy the triangle inequality and each tour starts and ends at the DC. Due to fixed working shifts, retailers usually define an aspired departure time from the depot (e.g., 5am) and maximum tour length (e.g., due to legal restrictions) for each tour and we therefore assume that vehicles depart from the DC at time zero and must return before time T, defined as the maximum tour duration. Waiting time between deliveries is not allowed as in practice drivers start unloading as soon as they arrive. A similar setting was also used by Kovacs et al. (2014b). Let V be the set of vehicles available for transportation at the DC. The number of vehicles available is assumed to be sufficiently large to fulfill the demand of all stores and consists of identical vehicles with a total capacity Q. As we consider MCVs, the loading area of each vehicle can be split into a limited number of compartments c. The number of compartments a vehicle may have active is indicated by $k \in K$, with $K = \{1, \ldots, c\}$. Due to the characteristics of an MCVRP that have been explained, loading and unloading costs are also decision-relevant costs in addition to the traveling costs. In line with this, l_k represents the loading cost of a vehicle dependent on the number of k compartments used, and u indicates the unloading cost of each stop (Hübner & Ostermeier, 2018).

The DC is responsible for the distribution of products from $|S|$ segments for a given planning horizon consisting of $|D|$ days, where S is the set of segments and D the set of days. It is assumed that the delivery pattern of each store for each segment (days in which a delivery should be made (Holzapfel et al., 2016)) are known, as well as the quantities to be delivered. q_{is}^{d} therefore defines the quantity of an order for segment s to be delivered to store $i \in N \backslash \{0\}$ on day d, and st_{is}^{d} represents the corresponding variable service time that depends on the delivery quantity. Additionally, a fixed service time sf is incurred each time a vehicle stops at a store, which is independent of the delivery quantity. Note that if day d does not belong to the delivery pattern of store i for segment s, q^d_{is} is set to 0.

Time windows are defined by the set $TW = \{1, \dots, tw\}$ and the intervals $[e_t, h_t]$, for every $t \in TW$, indicating the earliest (e_t) and latest (h_t) delivery time of time window t. The assignment of time windows will be made for each pair store-segment denoted as ((i, s): $i \in N \backslash \{0\}$, $s \in S$). Since not all time windows can be used for each pair store-segment (e.g., fresh products need to be received early), the subset $TW_{is} \subset TW$ indicates which time windows can be used for each pair (i, s). Since soft time windows are considered (see Section 2), a negative impact of having earlier or later arrivals than the time window assigned to a store have to be accounted. We follow Ioannou, Kritikos, and Prastacos (2003) and the ConVRP literature and impose unitary costs λ and β associated with one unit of time that the delivery is too early or late.

The objective of the PTWA-MCVRP is to minimize the total routing costs, considering traveling, loading and unloading costs, as well as the penalties for earlier or later deliveries than planned, while satisfying the stores orders. The problem combines a multi-period VRP for consistent deliveries with an MCVRP. As a consequence, it involves the following partial decisions that are made simultaneously and define the uniqueness of the problem formulation:

- Sequence of store visits, as in every VRP (this decision specifies the order in which each vehicle should perform the delivery of the orders assigned).
- Assignment of orders to tours/vehicles (this decision determines which segments are delivered simultaneously).
- Number and size of each compartment (this decision defines the number of compartments on each vehicle, and how the capacity is divided between compartments).
- Assignment of product-oriented time windows to each store-segment pair (this decision defines the arrival time of each segment at a store on each delivery day).

The first group of decisions is related to MCVRP. The last decision is associated with the consistent assignment of time windows along with penalty costs. We apply the following decision variables.

- $b^d_{ijv} = 1$ if vehicle v travels from location i to j on day d and $b^d_{ijv} = 0$ otherwise
- $\vartheta^d_{vs} = 1$ if vehicle v transports segment s on day d and $\vartheta^d_{vs} = 0$ otherwise
- $\theta^d_{isv} = 1$ if segment s is delivered by vehicle v to store i on day d and $\theta^d_{isv} = 0$ otherwise
- $r^d_{vk} = 1$ if vehicle v has k active compartments on day d and $r^d_{vk} = 0$ otherwise
- $y_{ist} = 1$ if store-segment pair (i, s) is assigned to time window t and $y_{ist} = 0$ otherwise

The continuous variables w^d_{iv} denote the arrival time at store i by vehicle v on day d and p^d_{is} the penalty cost incurred on day d by the pair (i, s) in the event of early or late deliveries. Additionally, the discrete auxiliary variables f^d_v represent the number of store stops performed by vehicle v on day d.

The PTWA-MCVRP can be formulated as follows:

$$\text{Minimize} \sum_{d\in D}\sum_{v\in V}\left[\sum_{k\in K} l_k \cdot r^d_{vk} + \sum_{i\in N}\sum_{j\in N} tc_{ij} \cdot b^d_{ijv} + u \cdot f^d_v\right] + \sum_{d\in D}\sum_{i\in N\backslash\{0\}}\sum_{s\in S} p^d_{is} \qquad (1)$$

subject to

$$\sum_{j\in N\backslash\{0\}} b^d_{0jv} \le 1 \qquad v \in V,\ d \in D \qquad (2)$$

$$\sum_{i\in N} b^d_{igv} = \sum_{j\in N} b^d_{gjv} \qquad v \in V,\ g \in N,\ d \in D \qquad (3)$$

$$\sum_{v\in V} \theta^d_{isv} \cdot M \ge q^d_{is} \qquad i \in N\backslash\{0\},\ s \in S, d \in D \qquad (4)$$

$$\sum_{s\in S} \theta^d_{jsv} \le |S| \sum_{i\in N} b^d_{ijv} \qquad v \in V,\ j \in N\backslash\{0\},\ d \in D \qquad (5)$$

$$\sum_{i\in N\backslash\{0\}}\sum_{s\in S} q^d_{is} \cdot \theta^d_{isv} \le Q \qquad v \in V,\ d \in D \qquad (6)$$

$$\sum_{i\in N\backslash\{0\}} \theta^d_{isv} \le M \cdot \vartheta^d_{vs} \qquad v \in V, s \in S, d \in D \qquad (7)$$

$$\sum_{s\in S} \vartheta^d_{vs} = \sum_{k\in K} k \cdot r^d_{vk} \qquad v \in V, d \in D, k \in K \qquad (8)$$

$$\sum_{k\in K} r^d_{vk} = 1 \qquad v \in V, d \in D \qquad (9)$$

$$f^d_v \ge \sum_{i\in N}\sum_{j\in N\backslash\{0\}} b^d_{ijv} \qquad v \in V, d \in D \qquad (10)$$

$$\sum_{t\in TW_{is}} y_{ist} = 1 \qquad i \in N\backslash\{0\},\ s \in S \qquad (11)$$

$$w^d_{0v} = 0 \qquad v \in V, d \in D \qquad (12)$$

$$w^d_{iv} \le M \cdot \sum_{s\in S} \theta^d_{isv} \qquad v \in V,\ i \in N,\ d \in D \qquad (13)$$

$$w^d_{iv} + tt_{ij} + sf + \sum_{s\in S} \theta^d_{isv} \cdot st^d_{is} - M \cdot \left(1 - b^d_{ijv}\right) \le w^d_{jv} \qquad v \in V,\ i \in N,\ j \in N\backslash\{0\},\ d \in D \qquad (14)$$

$$w^d_{iv} + tt_{ij} + sf + \sum_{s\in S} \theta^d_{isv} \cdot st^d_{is} + M \cdot \left(1 - b^d_{ijv}\right) \ge w^d_{jv} \qquad v \in V,\ i \in N,\ j \in N\backslash\{0\},\ d \in D \qquad (15)$$

$$w^d_{iv} + tt_{i0} + sf + \sum_{s\in S} \theta^d_{isv} \cdot st^d_{is} - w^d_{0v} - M \cdot \left(1 - b^d_{i0v}\right) \le T \qquad v \in V,\ i \in N\backslash\{0\},\ d \in D \qquad (16)$$

$$p^d_{is} \ge \left(\left(\sum_{t\in TW_{is}} y_{ist} \cdot e_t\right) - w^d_{iv}\right) \cdot \lambda - M \cdot (1 - \theta^d_{isv}) \qquad v \in V,\ i \in N,\ s \in S,\ d \in D \qquad (17)$$

$$p^d_{is} \ge \left(w^d_{iv} - \sum_{t\in TW_{is}} y_{ist} \cdot h_t\right) \cdot \beta - M \cdot \left(1 - \theta^d_{isv}\right) \qquad v \in V,\ i \in N,\ s \in S,\ d \in D \qquad (18)$$

$$
\begin{aligned}
&b^d_{ijv}, \vartheta^d_{vs}, \theta^d_{isv}, r^d_{vk}, y_{ist} \in \{0,1\}; && v \in V,\ \ i,j \in N,\ \ s \in S,\\
&w^d_{iv}, p^d_{is} \geq 0;\ \ f^d_v \in \mathbb{N} && d \in D, k \in K, t \in TW_{is} \qquad (19)
\end{aligned}
$$

Objective function (1) minimizes the total routing costs, including loading, traveling and unloading costs, plus the penalty costs of performing deliveries outside the bounds of the time windows assigned. The constraints of the problems can be aggregated into two groups. Constraints (2)–(10) compose the first group, which is related to the routing decisions of the MCVRP. Inequalities (2) and (3) ensure that each route starts at the DC, and that a store has only one predecessor and one successor on the route. Constraints (4) guarantee that a store receives all segments that it requires on each day. Constraints (5) ensure that store deliveries are only performed by a vehicle that actually visits the store. The vehicles' capacity is controlled by Constraints (6). Inequalities (7)–(9) define which segments are loaded on the vehicles and consequently how many compartments will be used. Constraints (10) determine the number of store stops each vehicle performs on a given day. The remaining constraints compose the second group and refer to the time window assignment. Constraints (11) ensure that only one time window can be assigned to each pair store-segment. These constraints ensure consistent deliveries and are hereafter denoted as *time window consistency constraint*. The departures from the DC at time zero are ensured by inequalities (12). The arrival times to stores are set by Constraints (13)–(15), ensuring that waiting time between deliveries is not allowed. Constraints (16) ensure that the tours do not exceed the maximum duration established. The penalty costs incurred by performing earlier or later deliveries than the bounds of the time window assigned are determined by Constraints (17) and (18). Note that the consistency requirement imposed by (11) is a hard constraint of our problem, while the time window satisfaction is hereby considered a soft constraint penalized in the objective function.

4. Solution approach

The main difficulty in solving the PTWA-MCVRP arises from the multi-compartment choice and routing, and in particular from the interrelation between the individual days of the planning horizon. In the problem considered, this interrelation relates to the consistent use of a unique time window for each customer-segment pair throughout the planning horizon. We propose an ALNS framework to cope with the characteristics of the problem regarding its two main groups of decisions: the routing problem with MCVs and the time window assignment.

ALNS algorithms are applied to different problem settings in literature and have been shown to provide good results for distinct VRP variants that are related to our setting, such as MCVRP (Derigs et al., 2011), VRPTW (Ropke & Pisinger, 2006) and ConVRP (Kovacs et al., 2014b). The ALNS framework was first introduced by Ropke and Pisinger (2006). Its central idea is to sequentially improve an initial solution by destroying and rebuilding parts of it. In the VRP variants, the destroying phase uses a destroy operator to remove a given number of requests from the routes, which are afterwards reinserted according to an insertion operator in the rebuilding phase. In an ALNS framework, several destroy and insertion operators are available and selected during the search procedure in an adaptive manner, depending on their performance during the search.

The ALNS framework developed to solve the PTWA-MCVRP combines daily and weekly operators to tackle the different problem decisions. The daily operators focus on a particular day and try to improve the routing decisions of the problem. This group of operators are usually proposed in the literature for VRPs. The weekly operators are new operators designed according to the characteristics of the problem. They have a broader scope, analyzing all the days at the same time, and aligning the time window assignment decisions. These operators are designed for our specific problem but can also be used to align other types of consistency (e.g., driver consistency) across the complete planning horizon. We therefore would like to note that the presented ALNS is capable of solving more general VRP variants such as time window assignment VRPs and consistency VRPs. The pseudo-code of the ALNS framework developed is shown in Algorithm 1. The general framework is explained below, and the main features are detailed in the subsequent subsections.

Algorithm 1 ALNS scheme for the product-oriented time window assignment for MCVRP.

```
generate a solution $S$                                          ▷ Section 4.1
set $S_{best} := S$
repeat
    select a destroy-repair heuristic pair $(d, r)$ based on adaptive
    weights $(\rho_{dr})$                                        ▷ Section 4.6
    if $d$ is a daily operator then                              ▷ Sections 4.2–4.4
        randomly select day $t$
        generate solution $S'$ by applying $(d, r)$ to $S$ on day $t$
    else                                                         ▷ Sections 4.2–4.4
        generate solution $S'$ by applying $(d, r)$ to $S$
    end if
    if $S'$ better than $S_{best}$ then                          ▷ Section 4.5
        $S_{best} = S'$
        $S := S'$
    else if $S'$ complies with the acceptance criteria then
    ▷ Section 4.5
        $S := S'$
    end if
    update performance of destroy-repair heuristic pair $(d, r)$
    ▷ Section 4.6
until maximum number of iterations is reached
return $S_{best}$
```

Although the problem formulation considers the time window consistency as a hard constraint, this is relaxed at the beginning of our solution approach. Actually, this constraint will be enforced during the search by adding an inconsistency cost to the objective function $(f(S'))$ as described in Eq. (20), creating a modified objective function $(f_a(S'))$, similar to Kovacs et al. (2014b).

$$f_a(S') = f(S') + \zeta \cdot inconPairs \qquad (20)$$

The inconsistency cost is set proportional to the number of customer-segment pairs with an inconsistent delivery plan (*inconPairs*), i.e., number of pairs with more than one time window assigned, and to the violation cost ζ. Parameter ζ is initialized at the beginning of the search and updated after a certain number of iterations $(\zeta = \exp^{(iterations/\delta)})$, with δ as control parameter for the increase of ζ. Naturally, the value of ζ and its increase has a high impact on runtime and solution quality as it decides how restrictive the search is concerning the consistency violation. An increasing ζ means that the more advanced we are in the search, the more costly it is to violate the *time window consistency constraint*. The inconsistency cost was introduced to allow a more diversified search for better routing options in the beginning of the search procedure as otherwise the routing would be very restricted. Please note that the inconsistency cost differs from the penalty costs as it is a relaxation only used to guide the search algorithm while the penalty costs for early/late deliveries are part of the soft time window constraints (Constraints (17) and (18)). The evaluation of our final solution is consequently based on the given MIP with objective function (1) and Constraint (11).

Table 1
Overview of destroy operators.

Type	Remove operators	Repair operators
Daily	Random, Shaw, worst	Greedy, regret
Weekly	Product-based, worst time window, worst arrival	

The algorithm starts with the generation of an initial solution S (see Section 4.1). In each iteration, a destroy-repair heuristic pair (d, r) is chosen by a roulette wheel selection, recurring to adaptive weights (see Section 4.6). The destroy operator can be selected from the group of daily or weekly operators (see Section 4.2). The removal step is followed by the reinsertion phase. The reinsertion is performed by the selected repair operator and each order is reinserted for the corresponding day on which the order is scheduled (see Section 4.3). After each remove and insertion, the arrival times of the orders are updated and the time window assignments are reset (see Section 4.4). We use six destroy operators and two different repair operators (see Table 1).

If the new solution S' meets the acceptance criteria, then it replaces S. If it improves the best solution found so far, according to the acceptance criteria, it replaces S_{best} (see Section 4.5).

4.1. Initial solution

The initial solution is generated by applying the savings heuristics of Clarke and Wright (1964) to each individual day of the planning horizon. This approach starts by creating routes with single orders and afterwards iteratively combines routes according to a calculated saving in traveling distance, while satisfying the vehicles capacity and maximum duration constraints. This heuristic is commonly used in different VRP problems, and was chosen because it provides a fast solution with a reasonable traveling distance. With the routes defined, the arrival times to each store are calculated ex-post, assuming the departures from the DC at time zero of each day. A time window is assigned afterwards to each individual order based on its arrival time, guaranteeing on-time deliveries. Note that at this stage the orders of each pair customer-segment can be assigned to distinct time windows. The first solution generated is therefore most probably not feasible with regard to the *time window consistency constraint*.

4.2. Destroy operators

In this solution approach, the destroy operators are separated into daily and weekly operators. Each of the operators was developed and tested to address a special characteristic of the problem. The daily destroy operators focus on a specific day of the planning horizon, and therefore focus on the routing decisions of the problem. The weekly destroy operators are the unique feature of our search procedure and were created to tackle the consistency aspect of the PTWA-MCVRP, thus focusing on the time window assignment decisions. In contrast to the daily operators, they analyze the entire planning horizon at once with the aim of aligning the time window assignment for the customer-segment pairs.

4.2.1. Daily operators

The daily operators remove r orders for a given day from its routes. The day selection is random, following a uniform distribution, but a day is set as “tabu” after its selection until all other days have also been selected, independently of the quality of the solution generated. The number of removes r is chosen randomly from the interval $[\kappa_1 \cdot N_{day}, \kappa_2 \cdot N_{day}]$, where N_{day} is the total number of orders to be delivered on the specific *day* and κ_1, κ_2 weights for the lower/upper bound.

The three daily destroy operators used are: *random removal*, *Shaw removal* and *worst removal*. These operators were proposed by Shaw (1997) and Ropke and Pisinger (2006) and are frequently used in the ALNS for different VRP variants. The general idea of each operator is given below.

The *random removal* operator randomly removes the orders from the set of routes of the day selected (using a uniform distribution). The *Shaw removal* removes the orders based on a similarity measure as denoted in Eq. (21). The similarity between two orders (z, m) is calculated based on four terms: distance, order size, arrival time and segment affiliation. These terms are weighted using the weights ϕ, ψ, φ and ω, respectively. The weights are used to balance the importance of the different terms for the search and therefore influence the solution quality. The terms with the highest impact should therefore be attributed with the highest weights. d_{zm} represents the distance between corresponding customers of the orders, $q(z)$ the order size, $a(z)$ the arrival time and s_{zm} the orders segment affiliation, i.e., $s_{zm} = 1$ if they are from same segment, 0 otherwise. The parameters d_{max}, q_{max} and a_{max} indicate the maximum distance between two customers and the maximum quantity and arrival time difference between any two orders across all available orders. The smaller R_{zm} gets, the more similar the orders are. In addition to the similarity measure, a randomization is used according to Shaw (1997) to diversify the search and ensure that not the most similar order is chosen. For this, a random number $z \in [0, 1)$ and a deterministic parameter λ is used. Based on the calculated similarity, the order that lies $z^{\lambda} \cdot 100$ percent down the similarity ranking is then chosen for removal.

$$R_{zm} := \phi \cdot \frac{d_{zm}}{d_{max}} + \psi \cdot \frac{|q(z) - q(m)|}{q_{max}} + \varphi \cdot \frac{|a(z) - a(m)|}{a_{max}} + \omega \cdot s_{zm} \tag{21}$$

Finally, the *worst removal* removes the orders that seem to be in a costly position in the solution. The cost of an order is the difference between the current solution cost and the solution cost if the order was removed (not having any additional cost of not being delivered). In this approach the solution cost is evaluated by the modified objective function (f_a). A randomized process controlled by the parameter λ is also integrated in this operator, similarly to Shaw removal, to ensure that it is not always the order with the worst cost that is removed.

4.2.2. Weekly operators

The weekly operators remove the orders of r customer-segment pairs from all days of the planning horizon. The number of removals r is chosen randomly from the interval $[2, \kappa_3 \cdot n]$, where n represents the number of customers and κ_3 the weight for the upper bound. Since all the orders of the r pairs are removed, the value of r has to be more restrictive than for the daily operators. By way of example, if two pairs are chosen with five orders for each pair throughout the planning horizon, this already results in ten orders for removal.

New destroy operators are designed in this work to specifically address the assignment decisions of the PTWA-MCVRP. Three weekly operators are developed: *product-based removal*, *worst time window removal*, *worst arrival removal*. The *product-based removal* is a variant of a random removal. The operator randomly selects r pairs of customer-segments (using a uniform distribution) and removes all orders of that pair from the solution, i.e., all orders of a segment that a customer placed in the planning horizon are removed. This operator diversifies the assignment of time windows. The other two operators are variants of a *worst removal* and are adaptations of the operators designed by Kovacs et al. (2015a). The *worst time window removal* operator calculates the number of time windows assigned to each customer-segment pair along the planning horizon and removes all the orders of the r pairs with the highest number of assignments. The aim of this operator is to reduce the number of time windows used for each pair and favor more consistent deliveries. The *worst arrival removal* considers the maximum arrival time difference between two orders of a customer-segment pair. It identifies the r pairs with the highest arrival time deviation and removes the two corresponding orders of each pair. It also aims at reducing the number of time windows but it is less disruptive since it does not remove all the orders from the same pair customer-segment.

4.3. Repair operators

Once a destroy operator is applied and orders are removed, the repair operator selected rebuilds the solution by reinserting the orders into the routes of their delivery days. If the removals were made by a daily removal operator, all the orders removed are from the identical day and the repair operator only considers that specific day. Otherwise, for weekly removals, the repair operator will focus on each day separately. From the list of days from which orders have been removed, a day is selected at random and all the orders of the corresponding day are reinserted according to the repair operator chosen. The process is repeated until all days are rebuilt.

Following most of the VRP literature that uses ALNS, two insertion heuristics are applied as repair operators: greedy insertion and regret insertions. These operators are based on Ropke and Pisinger (2006). The *greedy insertion* operator calculates the cheapest feasible position for reinsertion for each order removed, and the order with the lowest cost increase is selected to be inserted. The process is repeated until all orders are inserted. The *regret insertion* operators improve the greedy insertion by analyzing not just the best option for each order but the kth best, whereas k can have different values. This procedure integrates ahead information and calculates the regret of postponing an insertion. Let Δ_z^j denote the change in the objective value for inserting order z at its best feasible position on the jth cheapest route. The regret value is calculated according to Eq. (22) for all the orders removed. The order z with the highest regret value is selected to be inserted at its best feasible position. In each insertion, the regret value is recalculated for the set of orders remaining on the removal list until all orders are inserted.

$$regret_k^z := \sum_{j=2}^{k} \left(\Delta_z^j - \Delta_z^1\right) \tag{22}$$

4.4. Update of arrival times and time windows assignments

Since waiting time between deliveries is not allowed, every remove or insertion in a route impacts the arrival time of the successive orders and thus times need to be updated. All routes start at the DC at time 0 and therefore the arrival time of each order is calculated by consecutively adding the travel times between the customers visited (tt_{ij}) and their corresponding service time. As previously mentioned, the service time at a customer has a variable component proportional to the size of the orders delivered (st_{is}^d) and a fixed component per stop (sf, see also Section 2). Note that a customer can receive more than one order across different segments. In this case, the same arrival time is set for the corresponding customer orders and the service time at the customer is the total variable service time of the distinct orders plus the fixed service time.

Once the arrival times are updated, the new arrival time of an order might lie outside the bounds of the assigned time window. However, since the problem considers time window bounds as soft constraints, the solution is still feasible, but it yields a penalty cost for the deviation from the time window. The approach has flexibility to decide whether to change the time window assigned to the order, avoiding penalty costs which could cause inconsistent assignment, or to accept the penalty costs, maintaining the consistent assignment. It is therefore necessary to evaluate whether the time window assignment should be altered or maintained in order to achieve the minimal cost assignment, i.e., the best option between accepting a penalty cost or an additional inconsistency cost. Note that during the search, the time window assignment will be more restrictive as the constraint violation cost ζ increases.

The time window assignment update procedure is performed for each order separately, after the arrival time is determined. We define as $current_{cost}$ the current penalty and inconsistency cost induced by the pair customer-segment, disregarding the order being analyzed. This means we calculate the penalty and inconsistency costs for all orders of the customer-segment pair without the order that is currently under consideration. When deciding on the new time window assignment for the order being analyzed, there are two possible situations, as previously mentioned:

- Update only the time window assignment of the order being analyzed, maintaining the previous assignment of the remaining orders of the pair. This decision means that the total assignment cost of the pair customer-segment is calculated by adding the penalty and inconsistency costs associated with the time window assignment for the new order to the $current_{cost}$. The total cost of this situation is denoted as $singleUpdate_{cost}$, corresponding to a single assignment update.
- Update the assignment of all the orders from the pair customer-segment to the same time window, ensuring consistent deliveries throughout the planning horizon. This decision ensures the avoidance of any inconsistency cost (i.e., zero cost, as only one time window is assigned to the pair). The total assignment cost of the pair therefore comprises only the penalty costs that all the orders incur due to the new time window assigned. The total cost of this situation is denoted as $groupUpdate_{cost}$, and a new assignment is performed for the group of orders.

During the procedure, both costs $singleUpdate_{cost}$ and $groupUpdate_{cost}$ are calculated for each of the time windows available for the pair customer-segment (TW_{is}), and the cheapest assignment of all is chosen. The penalty cost that an order incurs is calculated by comparing the order arrival time with the bounds of the time window assigned. If the arrival time lies outside the time window bounds it causes a penalty cost proportional to the deviation. The inconsistency cost of updating the time window of a single order is determined by checking whether the time window assigned is already used by one of the other orders of the pair. The inconsistency cost is increased proportional to the current constraint violation cost (ζ) if the time window has not been used. We would like to note that the updating process is very time consuming and is also used to check each potential insertion position in the repair phase, therefore showing a high impact on the computational times. However, this updating process is required in our problem to align consistency of time windows across the planning horizon and as such represents one of the main features that characterizes our approach.

4.5. Acceptance criteria

The solution approach proposed uses a simulated annealing framework to evaluate and accept the solutions generated. A new solution S' is accepted as S_{best} if it improves the best solution. Otherwise, it is compared against the incumbent solution S by means of the probability $e^{-(f_a(S')-f_a(S))/\hat{t}}$. The parameter $\hat{t}$ denotes the current temperature. It is initialized at the beginning of the search such that a η% worse solution is accepted with a 50% probability

Table 2
Overview of numerical tests.

Section	Purpose	Data sets
5.2.1	Compare solution quality of our ALNS to TALNS of Kovacs et al. (2014b)	ConVRP
5.2.2	Analyze operators application frequency	ConVRP, retail data
5.2.3	Analyze algorithm performance	ConVRP, retail data
5.3.1	Evaluate impact of consistent deliveries	Retail data
5.3.2	Evaluate impact of product-oriented time window assignment	Retail data

and is decreased in every iteration with a cooling rate γ ($\hat{t} = \hat{t} \cdot \gamma$). Solutions are compared regarding the modified objective function f_a, allowing infeasible solutions to be accepted during the search. The ALNS algorithm stops after a given number of iterations.

4.6. Selection of a destroy-repair heuristic pair (d, r)

The pairwise selection of the destroy-repair heuristic pair is applied in each iteration and based on a roulette wheel selection principle, as proposed in Ropke and Pisinger (2006). The probability Φ_{dr} of a pair (d, r) being chosen is given by Eq. (23), where ρ_{dr} denotes the weight of the heuristic pair (Kovacs, Parragh, Doerner, & Hartl, 2012).

$$\Phi_{dr} := \frac{\rho_{dr}}{\sum_{d'=1}^{n_d} \sum_{r'=1}^{n_r} \rho_{d'r'}} \tag{23}$$

The weights ρ_{dr} are set to 1 at the beginning of the procedure, and updated dynamically during the search. Each pair (d, r) is associated with a score Ψ_{dr} that is updated each time the heuristic pair is applied according to the following criteria:

- $\Psi_{dr} + \sigma_1$, if the heuristic pair generates a new best solution;
- $\Psi_{dr} + \sigma_2$, if the heuristic pair generates a solution that has not been visited before, and is accepted as the new incumbent solution S.

As in Ropke and Pisinger (2006), the scores are initialized to zero and updated at each iteration according to the previous criteria. After a certain number of iterations, the weights ρ_{dr} are updated according to the recursive Eq. (24) and the scores are reset to zero for the next round.

$$\rho_{dr} := (1 - \alpha)\rho_{dr} + \alpha \frac{\Psi_{dr}}{max(1, \Theta_{dr})} \tag{24}$$

The parameter α is a reaction factor that controls how the weights are influenced by past and recent performances. In this way, it guides the search by controlling how sensitive the operator choice reacts to changes during the search.

5. Numerical experiments

Numerical experiments examine the effectiveness of the solution approach and the impact of the model extensions proposed. First, the performance of the ALNS is analyzed in Section 5.2, where solutions from related problem formulations from the literature are compared and an analysis of the operators execution is performed. Further analyses concerning the impact of our new model for grocery distribution are then presented in Section 5.3.

Table 2 provides an overview of the tests performed. The data sets used for these tests are described in Section 5.1.

The computational results presented in this section were obtained on a 3.60 gigahertz PC with a 16 gigabytes memory. The algorithm was implemented in C++ and run 10 times per instance in all the tests performed, stopping after 60,000 iterations. The constraint violation cost (ζ) and operators weights ρ_{dr} were updated every 100 iterations. Three regret insertion operators were used with $k \in \{2, 3, 4\}$. The remaining search parameters used are specified in Table 3. It provides an overview of the used parameters plus values, their function and how the used values were defined.

The choice of parameters influences both search quality and runtime. In particular, the κ-parameters show a high impact on both aspects as they control the number of removals dependent on the instance size. High values correspond to longer computational times, however they are required to guarantee a good solution quality. In this way we tuned the newly introduced κ_3 parameter (see Appendix B). Further, δ is responsible for the calibration of ζ, which is a key parameter of our algorithm. It dictates the violation cost and thereby influences the inconsistency cost and the search for consistent solutions. We therefore applied intensive studies to tune δ to fit our specific problem formulation (see Appendix A).

5.1. Overview of the data sets tested

In a first analysis, our problem and solution approach are compared to another VRP variant that considers consistent deliveries over multiple periods. This comparison is made to benchmark results provided by Kovacs et al. (2014b) for the ConVRP. It was chosen as it considers an identical approach with a departure time of zero at the DC. The other following analyses address our specific problem characteristics and are therefore performed on data based on grocery distribution. The data sets used in both cases are described in the following and summarized in Table 5.

5.1.1. ConVRP data set

To assess the effectiveness of our approach, we compare our solutions with ConVRP literature. These benchmark data do not deal with MCVRPs and hence are only a special case of our problem. The data sets used as benchmark were proposed by Groër et al. (2009) and Kovacs et al. (2014b), which were based on Christofides and Eilon (1969) instances for VRP considering a visit frequency of 70% (Groër et al., 2009), as well as 50% and 90% (Kovacs et al., 2014b). The visit frequency indicates the likelihood of a customer placing an order for each day in the planning horizon.

As our paper deals with a real-life problem in grocery distribution, we focus on the set of instances within the ConVRP data sets that consider a given maximum duration for tours and provided service times. In line with this, we tested nine instances with 50–100 customers to compare the results to the ones achieved by Kovacs et al. (2014b) with their template ALNS (TALNS, see Section 5.2.1).

5.1.2. Simulated data sets based on retail settings

Further data settings are based on direct information from a major European retailer. Loading and unloading costs have been derived following the insights from a preceding study by Hübner and Ostermeier (2018). The loading costs are presented in Table 4 and depend on the number of compartments per vehicle. Unloading costs accrue with every customer stop and are set to 2.20 currency units (CU). The transportation costs are based on the travel distance between any two locations i and j, $i, j \in N$. All experiments assume a vehicle capacity of 33 transportation units (TU).

Table 3
Search parameters setting.

Parameter	Function/Use	Value	Tuning/Origin
δ	Calibration of violation cost ζ	300	Own experiments (see Appendix A)
κ_1	Lower bound for removals (daily)	0.1	Pisinger and Ropke (2007) and Kovacs et al. (2015a)
κ_2	Upper bound for removals (daily)	0.4	Pisinger and Ropke (2007) and Kovacs et al. (2015a)
κ_3	Upper bound for removals (weekly)	0.1	Own experiments (see Appendix B)
ϕ	Weight Shaw removal	0.28	Derigs et al. (2011) and Hübner and Ostermeier (2018)
ψ	Weight Shaw removal	0.16	Derigs et al. (2011) and Hübner and Ostermeier (2018)
φ	Weight Shaw removal	0.28	Derigs et al. (2011) and Hübner and Ostermeier (2018)
ω	Weight Shaw removal	0.28	Derigs et al. (2011) and Hübner and Ostermeier (2018)
λ	Shaw/Worst parameter	4	Derigs et al. (2011) and Hübner and Ostermeier (2018)
σ_1	Update of score Ψ_{dr}	33	Ropke and Pisinger (2006)
σ_2	Update of score Ψ_{dr}	13	Ropke and Pisinger (2006)
α	Reaction factor ρ_{dr}	0.1	Ropke and Pisinger (2006)
η	Definition of starting temperature	0.05	Ropke and Pisinger (2006)
γ	Simulated annealing cooling rate	0.99975	Ropke and Pisinger (2006)

Table 4
Applied costs for loading MCV.

# Compartments	1	2	3	4
Loading (CU/shipping gate)	2.70	5.57	8.27	10.97

Further, early or late deliveries are penalized at 0.17 CU per minute, covering the representative labor cost of a store employee.

To further generalize our findings, we simulated 20 different instances with varying order and demand patterns[1]. The demand for each segment maps the representative order structure of the case company. The order structure has a visit frequency of about 70% (similar to that proposed by Groër et al., 2009). If a store is flagged up to be visited on a given day, it will place orders for all four available segments. Each instance comprises a planning horizon of seven delivery days and 50 stores served from a given DC. The individual order size for each segment is randomly chosen between a given minimum and maximum order quantity. The order quantity ranges between one and five TUs for the first segment and between one and ten TUs for the second segment. Furthermore, to map segments with a higher sales volume, segment 3 ranges between 5 and 20 TUs and finally segment 4 ranges between 10 and 25 TUs. The distance information is based on the VRP instances by Christofides and Eilon (1969). Note that the vehicle capacity in Christofides and Eilon (1969) is far higher than in our case example. However, if the distance data of Christofides and Eilon (1969) are multiplied by four to increase the travel distances (and keep the same density), one obtains comparable route lengths for grocery settings. All the settings of the generated instances are deduced from data provided by our partner from retail.

Table 5 summarizes the considered data sets for our tests and their characteristics.

5.2. Algorithm performance

In the following sections, we analyze the solution quality by comparing the results for the ConVRP benchmark instances, the application frequency of the ALNS operators proposed and the computational performance of the solution approach. The application frequency indicates if an operator was frequently called or not by the algorithm, regardless of the solution impact. Its value is calculated by dividing the number of times it was called by the number of iterations performed. In combination with the ALNS function (i.e., performance dependent weighing of operators) this enables insights on the use of each operator by the algorithm, as a frequently used operator corresponds to good scores and therefore a good performance.

5.2.1. Comparison of results with ConVRP benchmark instances

Instances modification. In contrast to our algorithm the TALNS does not consider the complete week for planning (i.e., no weekly operators) but constructs template routes for frequent customers and completes the daily planning afterwards. The total travel time plus service times (*TT*) for each instance are the comparison metric. Further, the problem addressed by Kovacs et al. (2014b) is a ConVRP that considers i) driver consistency and ii) arrival-time consistency. Their goal is to minimize traveling time while satisfying the two consistencies, i.e., to approach each customer using the same driver every day with a maximum arrival-time deviation (l_{max}) smaller than a pre-defined width *L*. These two aspects are not considered in our problem formulation and therefore the following modifications had to be applied to enable a fair comparison of both solution approaches.

[1] The generated instances can be found on http://www.vrp-rep.org/.

Table 5
Overview of data sets.

Data family, application(s)	Main characteristics	Comment
ConVRP, Comparison to TALNS (5.2.1), Operators analysis (5.2.2)	Planning horizon: 5 days Visit frequencies: 50, 70 and 90% Customers/Orders: 50, 75 and 100 Segments: 1 Number of instances: 9	– – One order per customer One product Instances with comparable setting by Kovacs et al. (2014b)
Retail, Operators analysis (5.2.2), Impact of consistent deliveries (5.3.1), Impact of product-oriented TW (5.3.2)	Planning horizon: 7 days Visit frequency: 70% Customers/Orders: 50/200 Segments: 4 Number of instances: 20	– – Multiple orders per customer Four incompatible product segments Simulated data based on retail information

Table 6
Results of comparison to TALNS best solution by Kovacs et al. (2014b).

# Customers	Visit frequency (%)	TALNS	Best ALNS	Avg ALNS	Best Gap (%)	Avg Gap (%)
50	50	2827	2865	2868	1.4	1.5
	70	4051	4050	4056	0.0	0.1
	90	4761	4762	4795	0.0	0.7
75	50	4702	4713	4792	0.2	1.9
	70	6786	6791	6880	0.1	1.4
	90	7749	7742	7881	−0.1	1.7
100	50	5349	5367	5416	0.3	1.3
	70	7209	7224	7298	0.2	1.2
	90	8745	8755	8840	0.1	1.1
Average	–	–	–	–	0.3	1.2

i) Driver consistency is integrated in our approach like the *time window consistency constraint*. This means that we relax the driver consistency constraint of Kovacs et al. (2014b) and introduce driver inconsistency costs by adding a violation cost to the modified objective function ($f_a(S')$) if multiple drivers (i.e., assignment to different tours) are used for the same customer. These costs are increased during the search, as explained in Section 4 for the time window consistency to enforce the driver consistency.

ii) Arrival-time consistency (l_{max}) is treated in the following way. The best l_{max} found for each instance by Kovacs et al. (2014b) is used as an input to define the width of our set of time windows. In this way we guarantee the arrival-time consistency achieved by Kovacs et al. (2014b) and can treat it like time window consistency during the search. Following this approach, the earliest time window given starts at time 0 with an l_{max} width and, from there, new time windows are available with a shift of one time-unit. Since Kovacs et al. (2014b) perform a set of tests with different lengths (L) for each instance, which result in distinct l_{max}, our comparison is made considering the l_{max} achieved for the tests with a maximum arrival time bound denoted as L_1. This limit is defined by the authors by running their algorithm without bounding the arrival time differences. Further, to ensure deliveries take place within the given time windows and thus adhere to l_{max}, we need to consider time windows as a hard constraint and therefore set the penalty costs for early/late deliveries to a very large number. Lastly, as only traveling times in test data are considered, loading and unloading costs are set to 0.

Comparison results. The numerical experiments confirm the ability of our algorithm to solve related problems effectively. Table 6 shows the TALNS solution and our ALNS best and average solutions, together with the corresponding gaps to our approach (Best Gap and Avg Gap, respectively).

In all tests, the solution approach was able produce consistent solutions in terms of driver and time windows, i.e., both driver and time window consistency constraints were satisfied. The results show that the proposed ALNS reaches solutions close to the TALNS, with the Best Gap close to 0%. The best solution reached by the ALNS was 1.4% worse only for the instance with 50 customers and a visit frequency of 50%. Additionally, for the remaining instances the average gap lies below 1.5% for most instances. These results demonstrate that our solution approach is able to find a consistent solution with a good traveling time. Please note that our approach was not developed to focus on driver consistency as in Kovacs et al. (2014b). Nevertheless, our algorithm provides promising results for this conVRP variant even if some further adaptions offer additional opportunities for future research.

5.2.2. ALNS operators analysis

Since we propose a new problem and developed an ALNS solution approach, an analysis of the operators considered is presented. We propose weekly destroy operators that are adjusted from the literature to cope with our problem characteristics. In this section, we compare the application frequency (AF) of each operator, i.e., the proportion of iterations each operator is called. This comparison is made for the runs with the ConVRP and the retail data sets.

Figs. 2 and 3 present the application frequency (AF) of the destroy operators (see Section 4.2) for both data sets; Figs. 4 and 5 present the same information, but for the repair operators (see Section 4.3).

When running our solution approach for the ConVRP data sets, results indicate that all the operators proposed are called during the search, with the daily operators having a share of 70%. Nevertheless, each of the weekly operators is also called around 10% of the times, with the product-based removal operator being the most called one due to the diversification that it allows.

The application frequency of these operators changes when the solution approach is used for the representative retail problem. Daily and weekly operators share an application frequency of 49%/51%, respectively. The results also indicate that the random and Shaw removals maintain an application frequency of around 20%, with an increase in calls for the product-based removal operator. The worst removal and worst arrival removal are called around 7% of the times across the tested instances. The weekly operators included in our solution approach clearly help the search for the PTWA-MCVRP.

Fig. 4 shows that for the ConVRP data set the four repair operators are similarly used by the ALNS, which is not the case for the retail data set. For the last, the greedy insertion operator has a much smaller application frequency than the remaining operators. Furthermore, the regret insertion with $k = 2$ seems to be the operator contributing the most, with an average application frequency of 40% (see Fig. 5).

5.2.3. Computational performance

Overall, we can confirm that the ALNS proposed is able to provide stable solutions for the PTWA-MCVRP due to a low variation in solution quality. The variation coefficient (standard deviation/ mean) was calculated over all test instances (10 runs per instance) within the given data sets. As a result, the average variation coefficient for our ALNS is 0.012 for the ConVRP data set and 0.003 for the retail data set. Furthermore, the solution approach is able to reach good quality solutions, improving the first feasible solution reached (respecting the formulation provided in Section 3) by an average around 16% (with minimums of 13% and maximums of 18% improvement, see Appendix D). The required computational time to solve practice-informed instances amounts to an average of one and a half hours. Since we consider a tactical planning problem, this constitutes an acceptable runtime effort. Nevertheless, we experimented with different iteration limits but found that a lower number of iterations leads to a decrease in solution quality. Additionally, a higher iteration limit increases the runtime significantly but does not have a real effect on solution quality (see Appendix C).

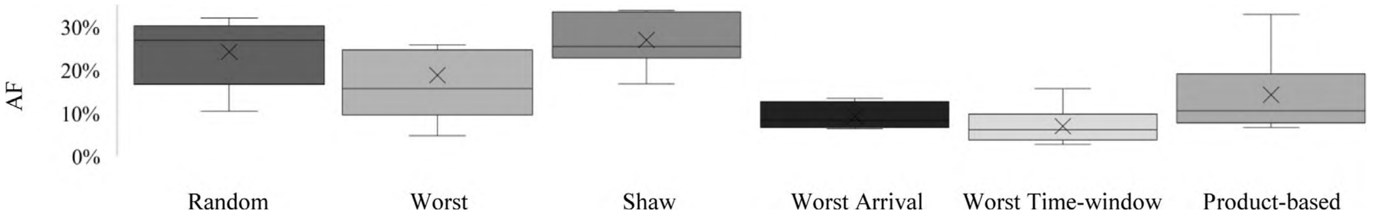

Fig. 2. Application frequency (AF) of destroy operators for ConVRP data sets.

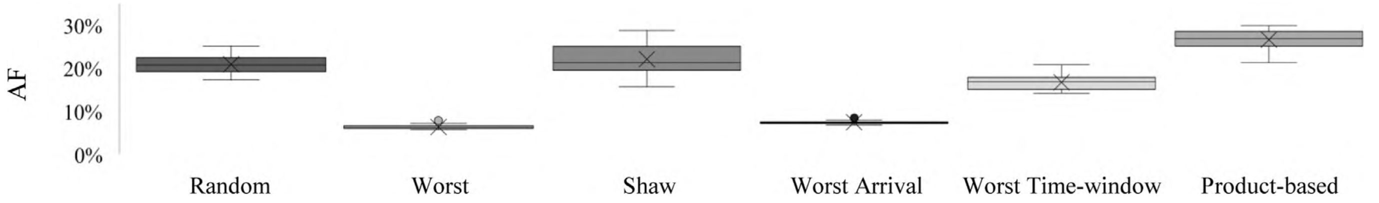

Fig. 3. Application frequency (AF) of destroy operators for retail data sets.

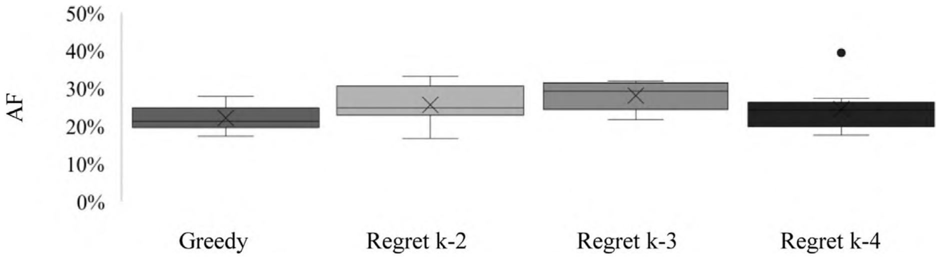

Fig. 4. Application frequency (AF) of repair operators for ConVRP data sets.

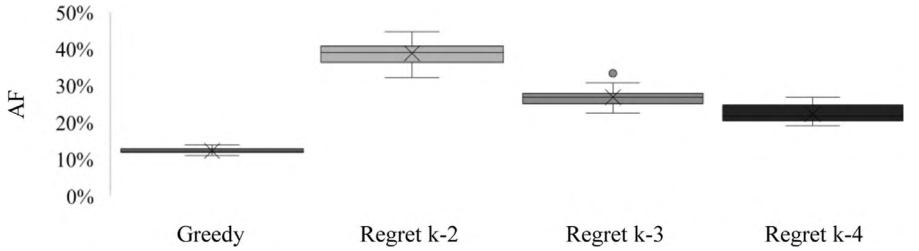

Fig. 5. Application frequency (AF) of repair operators for retail data sets.

The runtime required to solve the ConVRP instances is more variable, increasing proportional to the number of customers and visiting frequency. For the instances with 50 and 75 customers the runtime ranges from one hour for 50% visit frequency to four hours for 90% visit frequency. The computational effort for the 100 customer instances ranges from four to twenty hours. The data sets from the ConVRP require more time to solve because a much higher number of time windows need to be analyzed during the search. As the number of customers and time windows available increase, the update of arrival times and time windows assignment procedure (see Section 4.4) runtime increases. The check and update procedures of the arrival times and time windows assignment are responsible for the major part of computational time.

5.3. Impact analysis of time window assignment in grocery distribution

We further analyze the impact of introducing consistent deliveries and product-oriented time window assignment. We base our analysis on the retail data set. We start with the simplest case of introducing time windows assignment within the MCVRP with all time windows available and analyze the impact of consistent deliveries. Afterwards, the complexity is gradually increased by restricting the set of time windows for each pair customer-segment. In this way we can analyze the different characteristics of the problem.

5.3.1. Analysis of consistent time window deliveries

In this analysis we evaluate the impact of performing a consistent delivery planning, i.e., delivering each segment to stores within a unique time window assigned. We therefore compare our solution approach to a planning approach without consistency and an ex-post assignment of consistent time windows. In this experiment, a set of eight time windows with a one-hour width is considered, with the full set of time windows available to all customer-segment pairs. Two tests are performed for the comparison of each instance: (1) the ALNS is run for each day individually, considering only the daily operators with no time restrictions, and (2) the full ALNS is run for the complete planning horizon. The first test only attempts to minimize the routing costs, including the loading, traveling and unloading costs and does not consider consistent time windows. Dependent on the resulting arrival times of the orders, a time window is assigned (ex-post) to each customer-segment pair to enforce consistency. The corresponding time windows are chosen to minimize the penalty costs for early or late deliveries. The total cost in this scenario is calculated by summing up

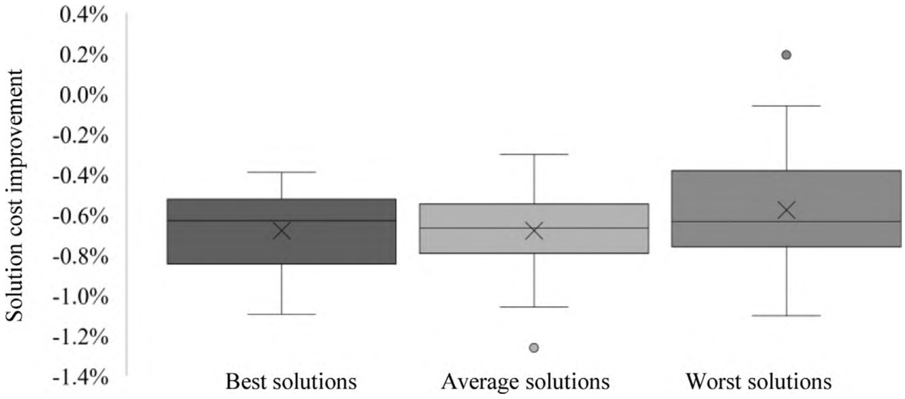

Fig. 6. Cost decrease of a solution generated by a consistent delivery planning compared to an unrestricted planning solution.

Table 7
Average percentage of orders delivered "X" minutes earlier or later than the time window assigned bounds.

Interval of time (X in minutes)	Unrestricted planning (%)	Consistent planning (%)
0	69	84
0–10	9	8
10–30	11	6
30–60	5	2
60–120	3	1
120–240	3	1
> 240	2	0

the routing costs of each day plus the penalty cost per customer-segment pair that arises to the ex-post assignment of consistent time windows. This first test is named *unrestricted planning* and is based on common practice. The second test aims at minimizing routing and penalty costs while ensuring consistent deliveries. The final solution provides a consistent time window assignment and therefore the test is referred to as *consistent delivery planning*. The improvement in total cost for the consistent delivery planning compared to the unrestricted planning is presented in Fig. 6 (improvements are indicated in negative %-values). Three cost scenarios are shown, namely the improvement in best, average and worst solution found in each test, for each instance. All tests provided solutions with a consistent time window assignment, i.e, for each customer-segment pair only one time window was assigned for the complete planning horizon.

The results show that consistent delivery planning enables better overall solutions than the unrestricted planning just focused on the routing. The best solution of the consistent planning is able to improve the unrestricted best solution by around 0.7%. Furthermore, the solutions generated with consistent planning are always better than unrestricted planning, with the cost deviation of the worst solution having an average improvement of 0.5%.

Although consistent planning provides solutions with better overall cost, the cost deviation between the two types of planning is low (below 1%). However, the solutions are very different. The routing cost of the consistent planning is between 1.1% and 1.7% higher than that for unrestricted planning, which is compensated by a 68% to 76% improvement of overall penalty costs. A further analysis of the delivery time of each order for both plans shows that consistent planning originates more on-time deliveries. Table 7 presents the average percentage of orders delivered "X" minutes outside the bounds of the assigned time window for both plans.

From the results of Table 7, we see that the consistent planning reduces the amount of deliveries performed outside the bounds of the time window and the amount of time deviation. While the unrestricted planning comprises 7% of the orders delivered with a deviation of at least one hour, the consistent planning reduces this percentage to 3%. Furthermore, note that the time deviations from the time window bounds are penalized in the overall cost of the solution by 0.17 CU per minute, covering the working cost of a store employee. However, the costs can be much higher for some of the deliveries in practice as for example spoilage and stockouts can occur additional to compensation payments for the violation of delivery times.

5.3.2. Analysis of product-oriented time window assignment

We further analyze the impact of defining a product-oriented time window assignment. In the previous tests, all pairs customer-segment had the full set of time windows available for assignment. However, in practice the stores might prefer to receive some segments in a more restricted set of time windows, as described in Section 2. We therefore tested three different scenarios:

1st scenario (Fresh TW):. Only a limited number of time windows can be used for one of the segments (representing the fresh products), with the full time window set available for the remaining segments. As the fresh products usually have to be delivered at the beginning of the day, and we assume departures at time 0, the set of time windows available for the fresh segment is set to the three earliest from the overall set.

2nd and 3rd scenarios:. Other segments might have additional restrictions in different stores, depending on their operations. We therefore used two random sets of time windows for the remaining segments. In the second scenario, named *Random TW (4–8)*, the number and time windows available are randomly selected between four and eight for each customer-segment pair. For the third scenario (*Random TW (4)*), the number of time windows available is fixed to four, and the time windows are randomly selected. In both scenarios, the set of time windows available for the fresh segment is the same as in the first scenario (*Fresh TW*).

The solutions obtained for the three scenarios are compared with the solutions from the consistent delivery planning with all

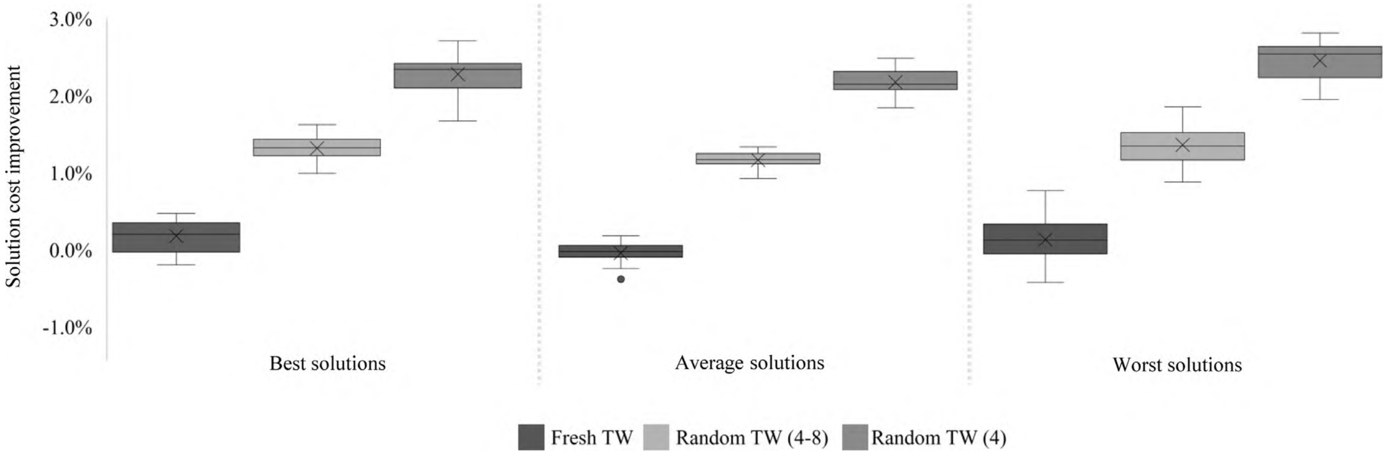

Fig. 7. Cost differences for solutions generated by a consistent delivery planning with distinct sets of time windows available compared to the solution with the full set available.

Table 8
Percentage of orders “X” minutes earlier or later than the time window assigned in each scenario.

Interval of time (X in minutes)	Unrestricted	All TW	Fresh TW	Random TW (4–8)	Random TW (4)
0	69%	84%	82%	72%	65%
0–10	9%	8%	9%	10%	10%
10–30	11%	6%	6%	11%	12%
30–60	5%	2%	2%	5%	8%
60–120	3%	1%	1%	2%	4%
120–240	3%	1%	1%	1%	1%
> 240	2%	0%	0%	0%	0%
Avg # TW per customer	1.32	1.32	1.42	1.92	2.22

time windows available (obtained in Section 5.3.1). The results are presented in Fig. 7 (improvements are indicated in negative %-values). Once again, the best, average and worst solutions achieved by each of the tests for each instance are compared.

The results show that the more restrictive the set of time windows available for each pair customer-segment is, the higher are overall costs of the solution. The costs therefore increase from the Fresh TW to the third scenario (Random(4)). The deviations between the best, average and worst solutions of each scenario are very similar. Analyzing the cost deviation of the best solutions for each scenario, we see that the *Fresh TW* scenario originates a small increase in costs (below 1%) in the solution cost compared with the full time window set available. This deviation reaches higher levels when all segments have random time windows available for assignment.

A further analysis of the two cost contributions (routing and penalty), indicates that the routing cost is very similar between all scenarios, with an average deviation of below 0.4%, pointing to the penalty cost as the main driver for the cost increase. Similarly to the previous section, Table 8 presents the average percentage of orders delivered “X” minutes outside bounds of the time window assigned for each scenario. The results from the previous unrestricted planning and consistent delivery planning analysis are also presented for comparison. The average number of time windows assigned per customer are presented for all scenarios in the bottom line.

The results show a reduction of on-time deliveries from the consistent scenario with all time windows available to the more restricted scenarios. We can see that when all time windows are available for all customer-segment pairs, the best solutions try to assign the same time window to the full range of products, having an average of 1.32 time windows assigned to each customer in both the unrestricted and consistent planning scenarios. As the scenarios constrain the time window set, the average number of time windows assigned per customer increases. This would lead to separate deliveries, which would increase the routing costs. However, it seems that by reducing the on-time deliveries, we maintain similar routing costs, leading to better overall costs.

6. Conclusion

This work extends the research on MCVRPs by addressing a multi-period setting with a product-oriented time window assignment. The resulting PTWA-MCVRP was studied for the grocery distribution application, which has particular characteristics due to the multiple products it distributes with distinct temperature requirements and the fact that it considers the impact of delivery plans on store operations. Notwithstanding, the proposed model and solution approach can also be transferred to other industries where MCVs and repetitive delivery cycles are applied (e.g., weekly glass waste collection, regular supply of petrol stations). The aim of the PTWA-MCVRP is to define a unique time window that should be used consistently throughout the planning horizon for each type of product of a store, taking into account the possibility of delivering the full product range jointly or separated with the use of MCVs. However, in practice the time windows are not hard constraints, and therefore deliveries outside the time window bounds are possible in practice with a negative impact for in-store operations. The objective of the problem proposed is thus to minimize the routing costs inherent in the use of MCVs and the penalty costs related to missing the time windows assigned.

An ALNS framework was designed to cope with the characteristics of the PTWA-MCVRP, combining daily and weekly operators to address the different problem decisions. The daily operators focus on a particular day and improve the routing, while the weekly operators have a broader scope, aligning the time window assignment decisions across all days. The algorithm was tested on benchmark instances for the ConVRP, which is closely related to our problem due to the arrival time consistency constraint, and generated instances based on a grocery distribution problem. We demonstrated that our solution approach provides very good results for the benchmark instances (ConVRP by Kovacs et al., 2014b) even though it was not developed to solve their given problem specifics. Furthermore, an analysis of the application frequency of the algorithmic operators showed that the introduced weekly operators effectively complement the ALNS framework with daily operators. They are frequently used during the search and were developed to allow a higher diversification for the PTWA-MCVRP.

An impact analysis of time window assignment in grocery distribution was also conducted. At a first stage, we showed that performing consistent delivery planning provides better overall solutions than unrestricted planning, just focused on the routing costs. Although the deviation cost achieved between both planning scenarios was small (0.7% average improvement), it was shown that there was a significant difference in the percentage of orders delivered outside the time window bounds, which could lead to higher costs due to spoilage, stockouts and fines. Finally, the implication of using a product-oriented time window assignment was analyzed by restricting the number of time windows available for assignment to the different products. We concluded that if all products have the same time windows available, most of the stores will receive the full range of products within the same time window. Therefore, as we restrict the set of time windows, the number of time windows used per customer increases, as well as the overall solution cost. This last effect is originated by the increase in the percentage of orders delivered outside the time window bounds, indicating that it is less costly to miss the time window than to change the routing.

Regarding the design of the algorithm proposed, we would like to emphasize that the newly introduced weekly operators were specifically designed to address the time window assignment decisions for repetitive deliveries throughout a period. Therefore, they can be used to solve VRP with time window assignment, even without the multi-compartment feature. This means that the general problem of MCVRP can be relaxed to the special case of single-compartment VRP and capacitated VRP. Further, it is able to solve other ConVRP variants, although it could require additional adaptations and algorithmic testing dependent on the specific problem setting (e.g., the consideration of driver consistency) to increase the efficiency in such settings. Further extensions of this work can be made by considering lower and upper bounds for violation of the time windows to prevent excessive penalties, as proposed by Ioannou et al. (2003). These bounds can be defined per customer and product, differentiating the cases that would not be so affected by the situation. For instance, stores with small backrooms or at high traffic locations would require more on-time deliveries than others. We assumed departures from DCs at time zero as these are typical in the retail context, and therefore a logical extension is to consider different departure times for the vehicles, as already considered by Kovacs et al. (2015b) for ConVRP. Additionally, considerations regarding the capacity of DC docks and fleet size could be included. Not all vehicles can depart at the same time due to loading dock capacity restrictions. Moreover, fleet size is dependent on the number of simultaneous deliveries, so having different departure times allows for a smaller fleet. The development of an exact approach, such as a branch-and-price, would also be a future research direction in order to achieve near optimal solutions for comparison and evaluate ALNS performance in greater detail.

A related extension of our work that mainly aims to solve the basic problem, is to further investigate different operators within the ALNS or even other algorithms to improve the computational efficiency and effectiveness.

Acknowledgments

The research was financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

The first author was also supported by Grant SFRH/BD/102013/2014 from FCT. The second author was funded by the German Ministry of Education and Research and the Hanns Seidel Foundation. The initialization of this international research cooperation was financially supported by the Bavarian Research Alliance.

Further, we would like to thank the anonymous reviewers and the Editor for their valuable recommendations, which have significantly improved our paper.

Appendix A. Algorithm tuning for δ

Table A1
Average deviation reached across all instances by changing δ.

	Avg. deviation[1] $\delta = 150$ vs. $\delta = 300$	Avg. deviation[1] Avg. $\delta = 300$ vs. $\delta = 600$
Best solutions reached	0.02%	−0.01%
Average solutions reached	0.00%	−0.01%
Average runtime	−1.02%	5.62%

[1] The average deviation shows the change in runtimes and solutions of increasing δ from 150 (300) to 300 (600); Example: the avg. runtime decreases by 1.02% if δ increases from 150 to 300.

Appendix B. Algorithm tuning for κ_3

Table B1
Average deviation reached across all instances by changing κ_3.

	Avg. deviation[2] $\kappa_3 = 5\%$ vs. $\kappa_3 = 10\%$	Avg. deviation[2] $\kappa_3 = 10\%$ vs $\kappa_3 = 20\%$	Avg. deviation[2] $\kappa_3 = 20\%$ vs $\kappa_3 = 30\%$
Best solutions reached	0.08%	0.02%	−0.06%
Average solutions reached	0.06%	0.08%	−0.11%
Average runtime	−2.22%	1.60%	12.53%

[2] The average deviation shows the change in runtimes and solutions of increasing κ_3 from 5 (10/20) to 10 (20/30); Example: the avg. runtime decreases by 2.22% if κ_3 increases from 5 to 10.

Appendix C. Algorithm tuning for iteration limits

Table C1
Impact of the number of iterations on the solution quality (best cost) and runtime (average).

Instance	Maximum number of iterations (limit)					
	30000		60000		90000	
	Solution	Runtime (in hour)	Solution	Runtime (in hour)	Solution	Runtime (in hour)
Retail 1	66070	0.60	65554	1.28	65394	2.14
Retail 2	71681	0.66	70800	1.40	70614	2.22
Retail 3	66055	0.62	65442	1.42	65300	2.14
Retail 4	68499	0.71	67791	1.54	67712	2.48
Retail 5	68366	0.67	67601	1.52	67430	2.48
Retail 6	66867	0.66	66370	1.35	66314	2.12
Retail 7	70718	0.70	70081	1.52	69978	2.44
Retail 8	70321	0.81	69646	1.60	69506	2.67
Retail 9	68019	0.69	67473	1.56	67340	2.36
Retail 10	69193	0.67	68534	1.48	68205	2.23
Retail 11	69411	0.79	68679	1.58	68476	2.74
Retail 12	68452	0.64	67711	1.34	67577	2.14
Retail 13	69626	0.68	68983	1.44	68866	2.29
Retail 14	68430	0.65	67766	1.40	67640	2.23
Retail 15	68684	0.71	68138	1.60	68048	2.34
Retail 16	70658	0.65	69902	1.43	69802	2.24
Retail 17	72084	0.68	71374	1.54	71280	2.54
Retail 18	68967	0.63	68492	1.37	68458	2.28
Retail 19	68177	0.70	67666	1.42	67509	2.22
Retail 20	69076	0.72	68443	1.59	68385	2.53
Avg. delta[3]	–	–	−645.4	0.79	−130.6	0.87
Avg. delta in %	–	–	−0.9%	115.7%	−0.2%	59.4%

[3] The average delta shows the absolute solution difference across all instances between the given limit and the corresponding lower limit (i.e., 30000 vs. 60000 and 60000 vs. 90000); Example: the runtime increases 0.79 hours (on average) if the limit increases from 30000 to 60000.

Appendix D. Solution improvement during search compared to initial feasible solution

Table D1
Average solution improvement after initial feasible solution.

Instance	Average improvement (%) (10 runs)
Retail 1	16.0
Retail 2	15.3
Retail 3	15.6
Retail 4	15.5
Retail 5	15.5
Retail 6	15.2
Retail 7	15.6
Retail 8	15.9
Retail 9	15.8
Retail 10	15.6
Retail 11	16.2
Retail 12	15.9
Retail 13	15.5
Retail 14	15.8
Retail 15	16.3
Retail 16	15.2
Retail 17	15.8
Retail 18	15.6
Retail 19	15.3
Retail 20	15.9
Average	15.7

References

Abdulkader, M. M., Gajpal, Y., & ElMekkawy, T. Y. (2015). Hybridized ant colony algorithm for the multi compartment vehicle routing problem. *Applied Soft Computing, 37*, 196–203.

Avella, P., Boccia, M., & Sforza, A. (2004). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research, 152*(1), 170–179.

Belhaiza, S., Hansen, P., & Laporte, G. (2014). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research, 52*, 269–281.

Chajakis, E. D., & Guignard, M. (2003). Scheduling deliveries in vehicles with multiple compartments. *Journal of Global Optimization, 26*(1), 43–78.

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society, 20*(3), 309–318.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Coelho, L. C., & Laporte, G. (2015). Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research, 242*(3), 854–864.

Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008). A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research, 191*(2), 295–305.

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

El Fallahi, A., Prins, C., & Calvo, R. W. (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research, 35*(5), 1725–1741.

Feillet, D., Garaix, T., Lehuédé, F., Péton, O., & Quadri, D. (2014). A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks, 63*(3), 211–224.

Groër, C., Golden, B., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & service operations management, 11*(4), 630–643.

Henke, T., Speranza, M. G., & Wäscher, G. (2015). The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research, 246*(3), 730–743.

Henke, T., Speranza, M. G., & Wäscher, G. (2017). A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working Paper No. 04/2017, Otto-von-Guericke-University Magdeburg.

Holzapfel, A., Hübner, A., Kuhn, H., & Sternbeck, M. G. (2016). Delivery pattern and transportation planning in grocery retailing. *European Journal of Operational Research, 252*(1), 54–68.

Hübner, A., & Ostermeier, M. (2018). A multi-compartment vehicle routing problem with loading and unloading costs. *Transportation Science*. doi:10.1287/trsc.2017.0775.

Hübner, A., & Schaal, K. (2017). An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effect. *European Journal of Operational Research, 261*, 302–326.

Hübner, A. H., Kuhn, H., & Sternbeck, M. G. (2013). Demand and supply chain planning in grocery retail: An operations planning framework. *International Journal of Retail & Distribution Management, 41*(7), 512–530.

Ioannou, G., Kritikos, M., & Prastacos, G. (2003). A problem generator-solver heuristic for vehicle routing with soft time windows. *Omega, 31*(1), 41–53.

Jabali, O., Leus, R., Van Woensel, T., & De Kok, T. (2015). Self-imposed time windows in vehicle routing problems. *OR Spectrum, 37*(2), 331–352.

Kaabi, H., & Jabeur, K. (2015). Hybrid algorithm for solving the multi-compartment vehicle routing problem with time windows and profit. In *Proceedings of the 12th international conference on informatics in control, automation and robotics (ICINCO): 1* (pp. 324–329). IEEE.

Kabcome, P., & Mouktonglang, T. (2015). Vehicle routing problem for multiple product types, compartments, and trips with soft time windows. *International Journal of Mathematics and Mathematical Sciences, 2015*, Article ID 126754, 9 pages. doi:10.1155/2015/126754.

Klingler, R., Hübner, A., & Kempcke, T. (2016). *End-to-end supply chain management in grocery retailing*. Cologne: European Retail Institute.

Koch, H., Henke, T., & Wäscher, G. (2016). A genetic algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working Paper No. 04/2016, Otto-von-Guericke-University Magdeburg.

Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014a). Vehicle routing problems in which consistency considerations are important: A survey. *Networks, 64*(3), 192–213.

Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2015a). The generalized consistent vehicle routing problem. *Transportation Science, 49*(4), 796–816.

Kovacs, A. A., Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2012). Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling, 15*(5), 579–600.

Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2014b). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks, 63*(1), 60–81.

Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2015b). The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research, 247*(2), 441–458.

Kuhn, H., & Sternbeck, M. G. (2013). Integrative retail logistics: An exploratory study. *Operations Management Research, 6*(1–2), 2–18.

Lian, K., Milburn, A. B., & Rardin, R. L. (2016). An improved multi-directional local search algorithm for the multi-objective consistent vehicle routing problem. *IIE Transactions, 48*(10), 975–992.

Martins, S., Amorim, P., & Almada-Lobo, B. (2017). Delivery mode planning for distribution to brick-and-mortar retail stores: discussion and literature review. *Flexible Services and Manufacturing Journal*, 1–28.

Muyldermans, L., & Pang, G. (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research, 206*(1), 93–103.

Ostermeier, M., & Hübner, A. (2018). Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research, 269*, 682–694.

Pires, M., Pratas, J., Liz, J., & Amorim, P. (2017). A framework for designing backroom areas in grocery stores. *International Journal of Retail & Distribution Management, 45*(3), 230–252.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*(8), 2403–2435.

Reed, M., Yiannakou, A., & Evering, R. (2014). An ant colony algorithm for the multi-compartment vehicle routing problem. *Applied Soft Computing, 15*, 169–176.

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

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

Spliet, R., Dabia, S., & van Woensel, T. (2017). The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science, 52*(2), 261–276.

Spliet, R., & Desaulniers, G. (2015). The discrete time window assignment vehicle routing problem. *European Journal of Operational Research, 244*(2), 379–391.

Spliet, R., & Gabor, A. F. (2014). The time window assignment vehicle routing problem. *Transportation Science, 49*(4), 721–731.

Subramanyam, A., & Gounaris, C. E. (2016). A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research, 248*(2), 384–395.

Subramanyam, A., & Gounaris, C. E. (2017). *Strategic allocation of time windows in vehicle routing problems under uncertainty*. Technical report, Carnegie Mellon University

Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., & Zhong, H. (2010). A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science, 44*(2), 193–205.

Tarantilis, C. D., Stavropoulou, F., & Repoussis, P. P. (2012). A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications, 39*(4), 4233–4239.

Taube, F., & Minner, S. (2018). Data-driven assignment of delivery patterns with handling effort considerations in retail. *Computers & Operations Research, 100*, 379–393.

van Zelst, S., van Donselaar, K., van Woensel, T., Broekmeulen, R., & Fransoo, J. (2009). Logistics drivers for shelf stacking in grocery retail stores: Potential for efficiency improvement. *International Journal of Production Economics, 121*(2), 620–632.