Journal of Optimization in Industrial Engineering

Vol.18, Issue 1, Winter & Spring 2025, 137-149

Research Article

Product-oriented split delivery in the multi-compartment vehicle routing problem

Mahnaz Shoeib [1], **Jafar Bagherinejad,**[2,*] **Mahdi Bashiri** [3]

1. Department of Industrial Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.
2. Department of Industrial Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.
3. Centre for Business in Society, Coventry University, Coventry, UK

https://doi.org/10.71720/joie.2025.1190026

Received: 10 October 2024
Revised: 20 January 2024
Accepted: 11 February 2025

Keywords:
Vehicle routing problem;
Multi-compartment vehicles;
Split delivery;
Matheuristic;
Adaptive large neighborhood search

Abstract

This paper addresses split delivery in the multi-compartment vehicle routing problem in which the possibility of splitting delivery depends on the product type. For certain product types, split delivery is not allowed but for other ones is permitted under a certain condition. The arrival time consistency is considered as split delivery condition. If the time difference between vehicle arrivals to the customer does not exceed a certain limit, the consistency of arrival time is established and split delivery is allowed. A mathematical model is developed to describe the proposed problem and used to solve small sized instances. To solve large sized instances, an adaptive large neighborhood search and a matheuristic based on fixing a part of customer to route assignment variables are developed. Computational experiments are performed on the multi-compartment vehicle routing problem with and without product-oriented split delivery. The effect of capacity and compartments per vehicle on delivery mode is investigated. The results demonstrate that the matheuristic outperforms the adaptive large neighborhood search in term of quality and computational time. Furthermore, delivery mode does not have a significant effect on the proposed algorithms' computational time and the number of vehicles is more affected by delivery mode than distance travelled.

Citation:
Shoeib, M., Bagherinejad, jafar, & Bashiri, M. (2025). Product-oriented split delivery in the multi-compartment vehicle routing problem. *Journal of Optimization in Industrial Engineering*, 18(1), 137-149. https://doi.org/10.71720/JOIE.2025.1190026

* **Corresponding Author:**
Jafar Bagherinejad
Department of Industrial Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.
E-Mail: **jbagheri@alzahra.ac.ir**

1. Introduction

Transportation is one of the most important parts of logistics systems. Since transportation costs comprise 10% of the product's total cost, saving transportation costs becomes a competitive factor (Doan et al., 2021). Previous studies have indicated how vehicle routing optimization can achieve significant economic savings (about 5-30% or 5-20%) (Cattaruzza et al., 2017). Therefore, the vehicle routing problem (VRP), introduced by Dantzig & Ramser (1959), has become one of the most studied optimization problems in operation research and logistics (Konstantakopoulos et al., 2022; Goli et al.,2018).

In the classical VRP, each customer is visited exactly once, split delivery is not allowed. In VRP with split delivery (VRPSD), the single-visit assumption is relaxed and each customer may be served by more than one vehicle. A customer's demand may be split due to one of these reasons: (1) the demand exceeds the vehicle capacity, in which case demand splitting is unavoidable. (2) Splitting demand can lead to significant cost savings (Toth & Vigo, 2014). By allowing split deliveries, the traveled distance, the number of vehicles, and their loading can be optimized (Kou et al., 2024; Han & Chu, 2016; Gu et al., 2024). Therefore, significant cost savings in logistics operations can be achieved by split deliveries. From the customers' viewpoint in VRPSD, a customer has to wait more than once to receive the total demand, which negatively affects customer satisfaction. Therefore, split delivery is often not suitable for products that customer prefers to receive all of them at once (Gulczynski et al., 2010).

Each customer may order several types of products that have to be transported separately due to their special transportation requirements, such as different temperature and humidity conditions. To distribute these incompatible products, multi-compartment vehicles (MCVs) are used which enable simultaneous transportation of the several non-mixable product types in different compartments (Guo et al., 2021). The multi-compartment vehicle routing problem (MCVRP) is a generalization of the classical VRP in which the vehicle capacity can be divided into several loading zones, i.e., compartments. Each compartment can be dedicated to only a single product type (Gu et al., 2024).

When demands of several product types are considered, split delivery is not that different product types can be delivered by different vehicles; it is the fulfillment of a demand for a single product type by more than one vehicle (Ostermeier et al., 2021; Henke, 2018). The benefits of split delivery depend on customer characteristics such as customers' locations and customers' demand patterns (Archetti et al., 2008).

Split delivery may only be allowed for certain product types and not for others (Alinaghian & Shokouhi, 2018).

For example, in companies such as FreshBox and Sysco, which distribute both perishable and non-perishable products, distribution strategies can vary depending on the product type. Customers prefer to receive perishable products at once, as repeatedly opening refrigeration units during multiple deliveries cause temperature fluctuations that accelerate spoilage. In contrast, split delivery is allowed in non-perishable products distribution.

Split delivery may only be beneficial under certain conditions, which to the best of our knowledge has not been studied so far.

This paper addresses the MCVRP with split delivery in which the possibility of splitting delivery depends on the product type. The split delivery of certain product types is not possible and these products must be delivered to the customer by only one vehicle. It is assumed that split delivery is allowed for other product types if the time interval between visits does not exceed a certain limit, which is called consistent arrival time. In other words, the distribution company can only achieve the benefits of splitting deliveries when it guarantees consistency of arrival time. Considering the NP-hardness of the proposed problem (Chen & Shi, 2019), a matheuristic and an adaptive large neighborhood search (ALNS) is designed to solve large-scale instances.

The main contributions of this study are as follows. First, we generalize an MCVRP by allowing product-oriented split delivery. The generalization is motivated by the fact that delivery modes depend on product type. For various reasons, such as ensuring inventory level or customer preferences, the split delivery of certain product types is not possible, while demand of some products may be split because of the dimensions, weight and availability of products. We propose an MIP model that can solve small sized instances to optimality. We also developed a matheuristic based on fixing a part of customer-to-route assignment variables. For comparison purpose, the performance of matheuristic is compared against ALNS.

The remainder of the paper is organized as follows. Section 2 briefly reviews the existing literature on MCVRPs. In Section 3, problem definition and mathematical model are presented. The proposed solution approaches are described in Section 4. The computational results are presented and discussed in Section 5. Finally, conclusions and suggestions for future studies are given in Section 6.

2. Literature Review

VRP, first introduced by Dantzig & Ramser (1959), is one of the most challenging optimization problems in the field of operation research, leading to extensive research in various conditions and application areas. Different variants of VRPs have appeared over the last decades (Elatar et al., 2023).

VRPSD is first introduced by Dror & Trudeau (1989), who demonstrated that considerable cost savings can be achieved by split delivery, both in terms of the total distance traveled and number of vehicles used. The VRPSD is shown to be NP-hard despite the relaxation of the single-visit assumption (Dror & Trudeau, 1990). Archetti et al. (2008) showed that the benefits of split delivery mainly depend on the characteristics of the demand. Archetti & Speranza (2012) provided a survey on the VRPSDs. Bortfeldt & Yi (2019) studied VRPSD and tree-dimensional loading constraints. They proposed a hybrid algorithm consisting of a local search algorithm for routing optimization, a genetic algorithm and several construction heuristics for packing. Lia et al. (2020) studied a VRPSD in which synchronization constraint, proportional service time and multiple time windows was

considered. They proposed a branch-price-and-cut algorithm to solve the problem. Wang et al. (2023) proposed a weighted open VRPSD with iterated clustering to simultaneously optimize the route of bus and passenger waking distance. To solve the proposed problem, they developed a max-min ant system algorithm by improving the decision mechanism for node access of vehicles. Zhang et al. (2024) studied VRPSD, multiple trips per vehicle and simultaneous pickup and delivery arising in the ground baggage handling problem at the airports. Other patricidal constraint such as time window, baggage release and back times were also taken into account. They developed ALNS using a two-stage solution evaluation method to solve this problem.

The aforementioned studies only consider the demand of one product type, which is delivered by single-compartment vehicles (SCVs). However, in practice, more than one product type may be requested (Chen & Shi, 2019). Gu et al. (2024) presented a survey on VRPs with multiple products in which the VRPs have been classified into two categories: VRP with compatible and incompatible products. The MCVRP, which was raised for the first time in fuel distribution, deals with the distribution of incompatible products. Application of the MCVRP include the perishable products distribution (J. Wang et al., 2023), waste collection (Mohammadi et al., 2023), fuel distribution (L. Wang et al., 2020), agricultural contexts (Polat & Topaloğlu, 2020) and etc.

Asawarungsaengkul et al. (2013) presented an MCVRP for distribution of liquid products in which customer demand is divided according to a predetermined pattern. The solution procedures including the optimization approaches (CPLEX), 2-opt algorithm, and clustering technique is proposed. Moshref-Javadi & Lee (2016) proposed an MCVRP with split delivery which focuses on the reduction of the customers' waiting time. A hybrid heuristic based on simulated annealing (SA) and variable neighborhood search (VNS) was designed to solve the problem. Urli & Kilby (2017) studied a multi-compartment fleet size and mix rich VRPSDs with time window and compatibility constraints arising in the context of fuel delivery. They suggested a constraint-based large neighborhood search (LNS) to solve the proposed problem. Alinaghian & Shokouhi (2018) introduced MCVRP with multi-depot in which split delivery is allowed only for a set of product types. They developed a hybrid solution approach that combines ALNS with VNS to solve large-scale instances. Zbib & Laporte (2020) developed a data-driven matheuristic for the commodity-split multi-compartment capacitated arc routing problem arising in recyclable waste collection. Wang et al. (2020) considered an MCVRP with split delivery and multiple trips in the context of fuel replenishment problem and proposed ALNS to solve this problem. Polat & Topaloğlu (2020) studied milk collection problem as MCVRP with split delivery and time limit in an uncertain environment. They implemented an enhanced iterative local search (EILS) algorithm to solve the proposed problem.

Although split delivery has received much attention in the literature, few studies have addressed MCVRP with split delivery. In the studied MCVRPs with split delivery, it is assumed that split delivery is always allowed and demand splitting does not depend on specific conditions.

3. Problem Description and Model Formulation

The proposed problem is defined on a complete undirected graph $G = (V,\ E)$ where $V = \{0\} \cup V'$ is the set of nodes and $E = \{(i,j) | i,j \in V, i \neq j\}$ is the set of edges. The depot is denoted by node 0 and the set $V' = V \setminus \{0\}$ contains all nodes corresponding to customers. A nonnegative travel time t_{ij} is assigned to each edge $(i,j) \in E$. It is assumed that all travel times are deterministic and symmetric ($t_{ij} = t_{ji}$).

Let $M = \{1, \dots, \mathcal{M}\}$ be a set of incompatible products that must be stored and transported separately. Each customer $i \in V'$ has a demand $d_{im} \geq 0$ for each product type $m \in M$. The depot has no demand ($d_{0m} = 0$). M_i' is a subset of products that must be delivered to the customer i at once. The unloading time of product m is denoted by tu_m.

Heterogeneous fleet $K = \{1, \dots, \mathcal{K}\}$ of vehicles is available at the depot, each equipped with a total capacity, Q_k. The total capacity of each vehicle $k \in K$ can be divided into a limited number of compartments, denoted by $L_k = \{1, \dots, \mathcal{L}_k\}$. The size of each compartment is chosen arbitrarily between 0 and Q_k in such a way that the total capacity of the compartments per vehicle does not exceed Q_k. Due to products incompatibility, each compartment of a vehicle is assigned to a single product type. The demand of each customer for each product type does not exceed Q_k. The vehicle fleet is assumed to be sufficiently large to satisfy all customer demands. Each vehicle is used for one tour at most. All vehicles depart from the depot at time 0 and have to return to the depot after visiting customers they serve. The transportation and unloading cost per time unit for vehicle k are denoted by ct_k and cu_k, respectively.

It is assumed that the possibility of splitting delivery depends on the product type. With respect to product type, two types of delivery mode are considered. A demand of customer $i \in V'$ for product type $m \in M_i'$ (certain product types for each customer) must be delivered by only one vehicle. In other words, split delivery of product $m \in M_i'$ is not allowed for customer $i \in V'$. However, split delivery of other product types ($m \in M \setminus M_i'$) is allowed if the time interval between visits does not exceed a certain limit (σ). The threshold reflects the maximum duration customers are willing to wait between receiving one portion of an order and the next. In this model, $\mathbb{M}$ is a sufficient large positive number. The objective of proposed problem is to determine routing plan that minimize total cost. To model the proposed problem, the following decision variables are defined:

x_{ijk} is a binary variable which equals to 1 if the edge $(i,j) \in E$ is travelled by vehicle $k \in K$, and 0 otherwise.

y_{imk} is a binary variable which equals to 1 if customer $i \in V$ receive product $m \in M$ from vehicle $k \in K$, and 0 otherwise.

w_{mlk} is a binary variable which equals to 1 if product $m \in M$ is assigned to compartment $l \in l_k$ of vehicle $k \in K$, and 0 otherwise.

$z_{imk} \in [0,1]$ is a non-negative continuous variable indicating the proportion of product $m \in M$ that is delivered by vehicle $k \in K$ to customer $i \in V'$.
$s_{ik} \geq 0$ is a non-negative continuous variable indicating the arrival time of vehicle $k \in K$ at customer $i \in V'$ and $s_{0k} = 0$. The mathematical model formulation is as follows:

$$Min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} \sum_{k \in K} x_{ijk} t_{ij} ct_k + \sum_{i \in V} \sum_{m \in M} \sum_{k \in K} z_{imk} d_{im} tu_m cu_k \quad (1)$$

$$\sum_{k \in K} z_{imk} = 1 \qquad \forall i \in V', m \in M, d_{im} > 0 \quad (2)$$

$$\sum_{j \in V, j \neq i} x_{ijk} \leq 1 \qquad \forall i \in V, k \in K \quad (3)$$

$$\sum_{i \in V'} \sum_{m \in M} z_{imk} d_{im} \leq Q_k \qquad \forall k \in K \quad (4)$$

$$z_{imk} \leq y_{imk} \qquad \forall i \in V', m \in M, k \in K \quad (5)$$

$$y_{imk} \leq \mathbb{M} z_{imk} \qquad \forall i \in V', m \in M, k \in K \quad (6)$$

$$y_{imk} \leq \sum_{l \in L_k} w_{mlk} \qquad \forall i \in V', m \in M, k \in K \quad (7)$$

$$\sum_{m \in M} w_{mlk} \leq \sum_{j \in V'} x_{0jk} \qquad \forall k \in K, l \in L_k \quad (8)$$

$$\sum_{j \in V, j \neq i} x_{ijk} = \sum_{j \in V, j \neq i} x_{jik} \qquad \forall i \in V, k \in K \quad (9)$$

$$s_{ik} + \sum_{m \in M} z_{imk} d_{im} tu_m + t_{ij} - \mathbb{M}(1 - x_{ijk}) \leq s_{jk} \qquad \forall i \in V, j \in V', k \in K \quad (10)$$

$$s_{ik} + \sum_{m \in M} z_{imk} d_{im} tu_m + t_{ij} + \mathbb{M}(1 - x_{ijk}) \geq s_{jk} \qquad \forall i \in V, j \in V', k \in K \quad (11)$$

$$s_{jk} \leq \mathbb{M} \sum_{m \in M} y_{jmk} \qquad \forall j \in V, k \in K \quad (12)$$

$$s_{0k} = 0 \qquad \forall k \in K \quad (13)$$

$$y_{jmk} \leq \sum_{i} x_{ijk} \qquad \forall j \in V, m \in M, k \in K \quad (14)$$

$$\sum_{k \in K} y_{jmk} \leq 1 \qquad \forall j \in V, m \in M_j' \quad (15)$$

$$\left| s_{jk} - s_{jk'} \right| \leq \sigma + \mathbb{M}(2 - y_{jmk} - y_{jmk'}) \qquad \forall k, k' \in K, k \neq k', j \in V', m \notin M_j' \quad (16)$$

$$\begin{aligned} &x_{ijk} \in \{0,1\} && \forall i, j \in V, k \in K \\ &y_{imk} \in \{0,1\} && \forall i \in V, m \in M, k \in K \\ &w_{mlk} \in \{0,1\} && \forall m \in M, l \in L_k, k \in K \\ &z_{imk} \in [0,1] && \forall i \in V, m \in M, k \in K \\ &s_{ik} \geq 0 && \forall i \in V, k \in K \end{aligned} \quad (17)$$

The objective function (1) minimizes the total cost including transportation and delivery costs. Constraints (2) ensure that each customer's demand for each product type must be satisfied. Constraints (3) guarantee that each vehicle performs at most one tour. Constraints (4) limit the vehicle capacity to Q_k. Constraints (5) and (6) link variables z_{imk} and y_{imk}. They set $y_{imk} = 1$ if a proportion of product $m \in M$ is delivered to customer $i \in V'$ by vehicle $k \in K$. Constraints (7) states that customer $i \in V'$ can receive product $m \in M$ from vehicle $k \in K$ if at least one compartment of this vehicle is assigned to the corresponding product. Constraints (8) ensure that each compartment of each vehicle leaving the depot is dedicated to only one product type. Constraints (9) is the flow conservation, generating that a vehicle that visits a certain node must leave it. Constraints (10) and (11) determine the vehicle arrival times to all customers in which waiting time between deliveries is not allowed. For customer $j \in V'$ who is not receiving any product from vehicle $k \in K$, constraints (12) set $s_{jk} = 0$. Constraints (13) guarantees that each vehicle departure from the depot at time zero. Constraints (14) link variables y_{jmk} and x_{ijk}. Constraints (15) ensure that each demand of product $m \in M_j'$ is delivered to customer $j \in V'$ by exactly one vehicle. Constraints (16) ensure that split delivery of product $m \in M \backslash M_j'$ is allowed for customer $j \in V'$ if the time interval between visits does not exceed a certain limit. Finally, constraints (17) define the domains of the decision variables.

4. Solution Approaches

4.1. Matheuristic

Matheuristics are hybrid optimization methods that make use of mathematical programming techniques in metaheuristics to the solution approach customization (Ngoo et al., 2024). To solve the problem, we propose a fix-and-optimize based matheuristic that iteratively selects a subset of variables to be fixed at their current values, while remaining sub problem is exactly or heuristically optimized (Dumez et al., 2023).
In the proposed matheuristic, three strategies are used to fix a part of the solution, all of which are based on fixing customers to route assignment variables. The type of fixing strategy is denoted by r ($r \in \{1,2,3\}$). In the first strategy, the demand of all product types ($m \in M$) of the certain customers is fixed in the routes. For each certain customer, in the second strategy only the demand of product type $m \in M_j'$ and in the third strategy only the demand of product type $m \in M \backslash M_j'$ is fixed in the routes. The pseudo-code of the developed matheuristic is given in Algorithm 1.
The algorithm starts with initializing the parameters (lines 1-3). The selection probability of each fixing strategy will depend on its weight. Initially, all fixing strategies have the same weight (line 4), i.e., the probability of selection is equal for all fixing strategies. The solution approach is initialized with a feasible initial solution (S_{ini}) obtained through a constructive heuristic (Section 4.3). The best solution (S_{best}) set equal to the initial solution (lines 5-6). A set of customers (V^*) is randomly selected (lines 7-8). The main loop of the algorithm is then started and repeated until a predetermined termination criterion is met (lines 9-29). In each iteration, a fixing strategy is chosen by the roulette wheel selection method (line 13). The selected customers ($i \in V^*$) are free to be assigned to any route while the assignment variables of other ones ($i \notin V^*$) are fixed at their current value according to the selected strategy (lines 14-19). After applying the

fixing strategy, the mathematical model (sub problem) is run with a short time limit TL (line 20). The best solution found is kept as an incumbent solution ($\acute{S}$). Then, the best solution and the weight of the selected strategy are updated. The weight of the selected strategy is increased by δ if it finds a new best solution. If S^* is not improved, the counter of non-improving iterations ($noImp$) is updated (lines 21-29) and other customers are selected randomly (lines 10-12). The algorithm stops when one of the predefined termination criteria is met. These criteria are the maximum number of iterations ($iter_{max}$) or maximum number of iterations without any improvement ($noImp_{max}$).

Algorithm 1 Matheuristic

1 Set parameters $iter_{max}$, $noImp_{max}$, $\acute{V}$, TL, M, $\acute{M_j}$
2 $noImp \leftarrow 0$
3 $iter \leftarrow 0$
4 Set initial weights of fixing strategy r equal to one ($w_r^0 = 1$).
5 Generate a feasible initial solution S_{ini}
6 $S_{best} \leftarrow S_{ini}$
7 $N \leftarrow$ generate a random number $\in [1, |\acute{V}|)$
8 $V^* \leftarrow$ Select N customers randomly as free customers.
9 **while** $iter < iter_{max}$ or $noImp < noImp_{max}$ **do**
10 **if** $noImp > 0$ **then**
11 $N \leftarrow$ generate a random number $\in [1, |\acute{V}|)$
12 Select N customers randomly as free customers
13 Select a fixing strategy using roulette wheel selection method based on previously weights.
14 **if** fixing strategy $r = 1$ is selected **then**
15 $y_{imk} = y^*_{imk}\ \ i \notin V^*, m \in M, k \in K$
16 **elif** fixing strategy $r = 2$ is selected **then**
17 $y_{imk} = y^*_{imk}\ \ i \notin V^*, m \in \acute{M_i}, k \in K$
18 **else**
19 $y_{imk} = y^*_{imk}\ \ i \notin V^*, m \in M \backslash \acute{M_i}, k \in K$
20 Run the model with TL and keep the best solution found so far ($\acute{S}$).
21 **if** $\acute{S}$ is better than S_{best} **then**
22 $noImp \leftarrow 0$
23 $iter \leftarrow iter + 1$
24 $S_{best} \leftarrow \acute{S}$
25 $w_r^{iter} \leftarrow w_r^{iter-1} + \delta$ (Update the weights of the fixing strategy)
26 **else**
27 $noImp \leftarrow noImp + 1$
28 $iter \leftarrow iter + 1$
29 **end while**

4.2. ALNS

The ALNS is a metaheuristic introduced by Ropke & Pisinger (2006) as an extension of LNS. Similar to LNS, ALNS is also based on the ruin and recreate principle. However, in contrast to LNS, several destroy and insertion operators are allowed to be used in the ALNS. The ALNS has been used successfully to solve different variants of the VRPs (Voigt, 2024). The pseudo-code of the ALNS is provided in Algorithm 2.

The ALNS starts with the construction of a feasible initial solution (S_{ini}) and initializing the relevant parameters (lines 1-4). The set of removal and insertion operators are denoted by Ω^- and Ω^+, respectively. All operators ($i \in \Omega^- \cup \Omega^+$) initially have the same weight (w_i) and the probability to be selected. All scores (π_i) are equal to zero (line 5). In the beginning of the algorithm, the best solution (S_{best}) and current solution (S^t) are equal to S_{ini} (line 6). The main loop of the algorithm is then started (lines 7-26) and repeated until a termination criterion is met. In each iteration, a removal and insertion operators are selected using the roulette wheel selection (line 8). The selected removal and insertion operators are successively applied to destroy the current solution and then repair it to generate a new solution $\acute{S}$ (line 11). The new solution is accepted as the current solution depending on an acceptance criterion (Section 4.2.1). Then, the best solution is updated. If S_{best} is not improved, the counter of non-improving iterations ($noImp$) is updated and another removal and insertion operators are selected using roulette wheel selection based on previously obtained scores (lines 9-10). For every γ iterations, the weight of each operator is updated according to their performance (lines 24-25) (Section 4.2.3). The algorithm terminates when either a maximum number of iterations ($iter_{max}$) or maximum number of iterations without improvement ($noImp_{max}$) has been reached.

4.2.1. Acceptance criterion

The acceptance criterions are implemented in different ways to decide whether a new solution should become the current solution or not. The simplest acceptance criterion is to only accept improving solution. In the proposed ALNS, acceptance criterion is based on SA, in which a new solution with objective value $Obj(\acute{S})$ is always accepted if it improves the current solution with objective value $Obj(S^t)$, i.e., $Obj(\acute{S}) < Obj(S^t)$. Otherwise, a new solution is accepted with probability of $e^{-(\frac{Obj(\acute{S}) - Obj(S^t)}{T})}$. $T > 0$ is the current

temperature. Initially, T is set to a value that non-improving solution are always accepted. Then, it is decreased by a cooling factor (τ) in each iteration where $0 < \tau < 1$ (Zhang et al., 2024)

Algorithm 2 ALNS
Generate a feasible initial solution, S_{ini}
Set parameters $iter_{max}$, $noImp_{max}$
$iter \leftarrow 0$
$noImp \leftarrow 0$
Set $w_i = 1$ and $\pi_i = 0$ for each operator ($i \in \Omega^- \cup \Omega^+$)
$S_{best} \leftarrow S^t \leftarrow S_{ini}$
while $iter < iter_{max}$ or $noImp < noImp_{max}$ **do**
  Select a removal and insertion operator using roulette wheel selection based on previously obtained scores
  if $noImp > 0$ **then**
    Select a removal and insertion operator using roulette wheel selection based on previously obtained scores
  $S' \leftarrow$ Apply selected removal and insertion operators to generate a new solution
  if S' is accepted **then**
    $S^t \leftarrow S'$
    if S' is better than S_{best} **then**
      $S_{best} \leftarrow S'$
      $noImp \leftarrow 0$
      $iter \leftarrow iter + 1$
    else
      $iter \leftarrow iter + 1$
      $noImp \leftarrow noImp + 1$
  else
    $iter \leftarrow iter + 1$
    $noImp \leftarrow noImp + 1$
  if mod $(iter, \gamma) = 0$ **then**
    Update the weights of operators and reset scores
end while

4.2.2. Removal and reinsert operations

The removal operator removes a number of customers (p) from the current solution and the insertion operator reinserts the removed customers. To ensure feasibility, first all customers who are visited more than once are selected, then the number of customers who are visited only once are selected based on the removal operator. The following operators are used in the proposed ALNS.

Random removal: The random removal operator selects ρ customers randomly and removes them from the current solution.

Vehicle removal (Route destruction): A vehicle k is randomly selected. The customers served by vehicle k are selected until the number of customers is equal to p. If all customers served by vehicle k have been selected and the number of selected customers is less than p, another vehicle is selected. This process continues until p customers are selected (Mancini, 2016).

Shaw removal: The idea of removing customers based on their similarities was first proposed by Shaw (1998). The first customer is chosen randomly. In the subsequent iterations, Shaw removal removes the customer that is most similar to the customer removed in the preceding iteration. The procedure is repeated until p customers are removed. In this paper, the similarity between two customers i and j is expressed by R_{ij} based on travel time and demand by Eq. (18).

$$R_{ij} = \varphi \frac{t_{ij}}{t_{max}} + (1-\varphi) \frac{\left|\sum_m d_{im} - \sum_m d_{jm}\right|}{d_{max}} \qquad (18)$$

Where t_{max} indicate the maximum travel time between any pair of customers and d_{max} is the maximum demand $(\sum_m d_{im})$. $\varphi \in [0,1]$ is a relative weight of each term.

Greedy insertion: For each removed customer (i.e., unassigned customer), all feasible positions in the fleet are determined. The customer with the lowest insertion cost is then selected and inserted in its lowest-cost position. Then, the insertion costs of the remaining customers are updated and this procedure is repeated until all customers have been reinserted.

Random insertion: For each removed customer, all feasible positions in the fleet are determined. A customer is selected randomly and inserted in its random feasible position. The feasible positions of the remaining customers are determined and this procedure is repeated until all customers have been reinserted.

4.2.3. Adaptive weight adjustment

For every γ iteration (i.e., time segment j), the weight of each operator is updated according to their performance by Eq. (19).

$$w_{ij+1} = (1-\lambda) w_{ij} + \lambda \frac{\pi_i}{\theta_i} \qquad (19)$$

Where $\lambda \in [0,1]$ is a parameter that controls how the weights are influenced by the historical performance. θ_i is the number of times that operator i was called in the last γ iteration. π_i is the current score of the operator i. The parameters v_1, v_2 and v_3 are used to update the operator' score by Eq. (20), where $v_1 > v_2 > v_3$ (Friedrich & Elbert, 2022).

$$\pi_i^{\gamma+1} = \pi_i^{\gamma} + \begin{cases} v_1 & \text{If new best solution has been found.} \\ v_2 & \text{If new solution is accepted but worse than best solution.} \\ v_3 & \text{If new solution is rejected.} \end{cases} \qquad (20)$$

4.3. Initial Solution

A feasible initial solution is generated through a constructive algorithm. If the customer's demand exceeds the capacity of

a vehicle, it is splitting into several customers, each of which requests only one product type. With a such splitting, each customer can be assigned to each vehicle. First, a customer is randomly selected and assigned to a randomly selected vehicle to generate a feasible route. Then, unassigned customers are inserted by Greedy insertion heuristic (Section 4.2.2). If insertion is not possible anymore due to capacity constraints, a new vehicle is selected and this procedure is repeated until all customers have been inserted.

5. Numerical experiments

This section presents numerical experiments to investigate the efficiency of the proposed model and solution approaches. The experiments are implemented in Python, utilizing Gurobi optimizer, version 10.0.3 as the optimization solver with a time limit of 14400 seconds. All computational experiments are performed on a laptop with 16 GB RAM and 2.8 GHz Intel Corei5-3210M processor running Windows 10.

5.1. Test instances

The computational results are based on instances derived from Martins et al. (2019), which includes 50 customers who order incompatible products in several time periods. In this data set, the capacity of all vehicle is identical, at 33 units and the time required to unload each product type is 2 units ($tu_m = 2$).

We generate two sets of small and large sized instances based on Martins et al. (2019). The small sized instances are generated by randomly extracting up to 15 customers with up to 3 product types from each instances in Martins et al. (2019). On the other hand, 20 to 50 customers with up to 3 product types are extracted as large sized instances. For each combination of customer and product type, 5 instances are generated (60 instances in total) which is indicated in column "EX" in detailed Table A1 in Appendix. The proposed problem is a single-period MCVRP in which the arrival time consistency of several vehicles is studied; therefore, the computational experiments are conducted over a single period (The first period was randomly selected.).

It is assumed that the fleet is sufficiently large to serve all demand. The maximum number of compartments per vehicle is equal to the number of products, i.e., $\mathcal{M} = \mathcal{L}_k$ for $k \in K$. Transportation and unloading costs of vehicles per time unit are set to 10 and 3.75 units, respectively. Each instance was solved with and without considering customer requirement in term of splitting delivery (product-oriented split delivery or general split delivery). These instances are described as $n - \mathcal{M} - Gsplit/Psplit$, where n is the maximum number of customers, $\mathcal{M}$ is the number of product types and the last field $Gsplit/Psplit$ refers to customer requirement in term of splitting delivery. In the case of product-oriented split delivery ($Psplit$), the demand of each customer for given product types cannot be split ($m \in M_i^{'}$ for $i \in V^{'}$), but split delivery of other product types is allowed under certain condition. In $Gsplit$, split delivery is allowed for all product types. It is assumed that the demand of first product type cannot be split in product-oriented split delivery mode ($M_i^{'} = \{1\}$ for $i \in V^{'}$).

The maximum allowed arrival time difference is assumed to be 1 hour. Note that the proposed model contains a large positive number $\mathbb{M}$. The upper bound of $\mathbb{M}$ can be calculated by Eq. (21).

$$\mathbb{M} \geq \sum_{i \in V} \sum_{j \in V^{'}} t_{ij} + \sum_{j \in V^{'}} \sum_{m \in M} d_{jm} tu_m \qquad (21)$$

Since the proposed solution approaches has a random component, each algorithm has been run 5 times on each instance. The best and average objective function value over 5 runs, denoted as Obj_{best} and Obj_{avg} respectively, are reported. The average computational times, denoted as T_{avg}, are presented. The adaptive weights are updated every 5 iterations. All parameters are tuned experimentally to achieve a reasonable balance between effectiveness and computational time. An overview of the parameters used in the solution approaches is provided in Tables 1 and 2.

Table 1
The parameters of matheuristic

Notation	Description	Value
$iter_{max}$	Maximum number of iterations	50
$noImp_{max}$	Maximum number of non-improving iterations	5
TL	Time limit (in seconds)	10
δ	Score used in updating the weight of fixing strategy	0.1

Table 2
The parameters of ALNS

Notation	Description	Value
$iter_{max}$	Maximum number of iterations	300
$noImp_{max}$	Maximum number of non-improving iterations	100
T	Initial temperature in SA	0.4
τ	Cooling factor in SA	0.9994
φ	Shaw parameter	0.5
γ	Number of iterations per segment	5
λ	The parameter is used to weight adjustment	0.3
v_1	The parameter is used to update the operator' score if a new best solution is found.	20
v_2	The parameter is used to update the operator' score if a new solution is accepted.	10
v_3	The parameter is used to update the operator' score if a new solution is rejected.	1

5.2. Computational results

This section presents the computational results using the solution approaches described in Section 4. We solve the small sized instances with ALNS and matheuristic and compare the results with optimal solution obtained by Gurobi. Table A1 in Appendix presents the computational results of the small sized instances in detail. The large sized instances are solved by ALNS and matheuristic, since Gurobi is not able to find a solution when the number of customers or product types grows up to 20 and 3, respectively. Tables 3 and 4 present the average computational results of the small and large sized instances, where each row represents the average results of five instances with the same combination of customer and product.

These tables are organized as follows. First column indicates the name of instance, which is composed of the number of customers, the number of products, and finally splitting mode ($Gsplit/Psplit$). Under the Gurobi header, optimal objective function value (Obj^*), computational time (T) and optimality gap (Gap^*) are reported. The optimality gap is the percentage gap between Obj^* and lower bound reported by Gurobi. Under the ALNS and matheuristic headers, the best and average objective function value over five runs is reported, denoted as Obj_{min} and $\overline{Obj}$ respectively. The corresponding computational times are denoted as T_{min} and $\bar{T}$ respectively. The relative gap ($\overline{Gap}$) is measured with respect to Gurobi given by $(\overline{Obj} - Obj^*)/Obj^*$. The relative improvement of the matheuristic compared to ALNS is also reported in column "$\overline{Imp}$", which is the percentage gap between $\overline{Obj}$ in matheuristic and $\overline{Obj}$ in ALNS. The average results over all instances are given in the last row of each table.

As illustrated in Table 3, Gurobi only solves instances up to 15 customers and 2 product types within a time limit of 4 hours. As shown in Table A1 in Appendix, in 12 out of 15 instances, Gurobi found the same optimal solution in both splitting delivery modes ($Gsplit/Psplit$). This means that in the most small sized instances it is possible to meet customer requirements in term of demand splitting without additional costs. In the remaining three instances, the proposed model in the product-oriented split delivery mode costs more than the general mode.

As the problem size increase, the computational time of Gurobi increases exponentially. In all small sized instances, both proposed algorithms find near optimal solutions within very short computational times. The ALNS is able to find high quality solutions with an average relative gap of 0.5% in less than 10 seconds per small sized instance. The matheuristic obtains near optimal solution with an average gap of only 0.12% within an average run time of 25.6 seconds. The proposed matheuristic on an average achieve improvement of 0.38%, when compared to the proposed ALNS. There are no significant differences in computational times across the two algorithms.

Note that although the Gurobi computational time is affected by the mode of demand splitting, it does not have a significant effect on the proposed algorithms computational time.

In Table 4, we evaluate the performance of the matheuristic by comparing its results to the results of ALNS in the large sized instances. Table 4 shows that although the size of problem increases, both algorithms can obtain a solution in relatively short time. Based on the computational results provided in Table 4, the proposed matheuristic perform better than ALNS on all instances with an average improvement of 7.7%. Matheuristic are efficient, reducing up to nearly 20% ((141.9-178.2)/178.2) of computational time on average. Allowing product-oriented split delivery mode in MCVRP may lead to an increase in the number of vehicles needed or affect routing decisions and travelled distance. Therefore, as expected, the route generated in product-oriented mode are costly than the general mode in the large sized instances.

Table 3
Small sized instances

	Gurobi			ALNS					Matheuristic					
Instance	Obj^*	Gap^* (%)	T (s)	Obj_{min}	$\overline{Obj}$	$\overline{Gap}$ (%)	T_{min} (s)	$\bar{T}$ (s)	Obj_{min}	$\overline{Obj}$	$\overline{Gap}$ (%)	T_{min} (s)	$\bar{T}$ (s)	$\overline{Imp}$ (%)
10-2- $Gsplit$	12488.4	0.00	138.3	12488.4	12552.7	0.43	7.4	7.3	12488.4	12488.7	0.00	17.8	20.4	-0.43
10-2- $Psplit$	12488.4	0.00	54.2	12488.4	12488.4	0.00	4.7	5.7	12490.1	12490.1	0.01	8.9	9.1	0.01
10-3- $Gsplit$	15196.5	0.00	436.7	15253.7	15288.5	0.54	12.0	12.0	15196.5	15217.9	0.13	21.2	26.0	-0.40
10-3- $Psplit$	15196.5	0.00	749.1	15196.5	15263.9	0.40	5.7	6.8	15196.5	15216.2	0.11	16.8	21.8	-0.29
15-2- $Gsplit$	18484.6	0.00	8993.3	18484.6	18631	0.79	13.5	14.3	18484.6	18514.5	0.17	38.2	44.6	-0.62
15-2- $Psplit$	18597.5	0.00	4909.9	18669.6	18750.9	0.82	9.7	11.3	18615.7	18651	0.27	27.4	31.9	-0.54
average		0.00	2546.9			0.50	8.8	9.6			0.12	21.7	25.6	-0.38

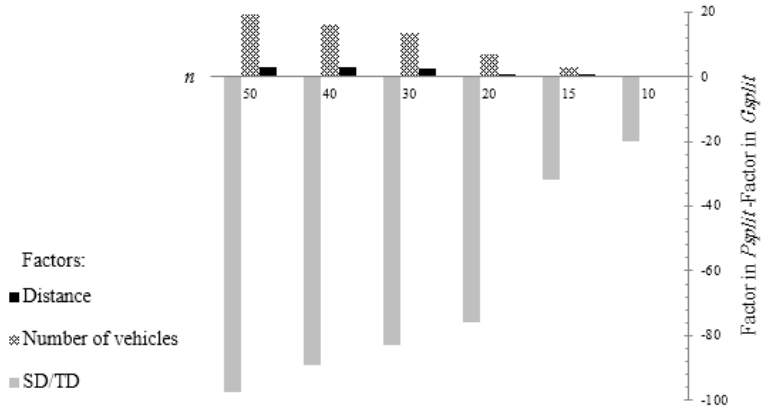

Fig.1 Comparison of MCVRP in two modes: $Gsplit$ and $Psplit$

Table 4
Large sized instances

Instance	ALNS				Matheuristic				$\overline{Imp}$ (%)
	Obj_{min}	$\overline{Obj}$	T_{min} (s)	$\bar{T}$ (s)	Obj_{min}	$\overline{Obj}$	T_{min} (s)	$\bar{T}$ (s)	
20-2- $Gsplit$	28527.9	29192.5	50.0	48.4	27915.9	28163.0	125.8	114.4	-3.53
20-2- $Psplit$	28921.0	29380.9	33.7	35.0	28134.6	28295.1	56.6	52.4	-3.70
30-2- $Gsplit$	47471.1	49265.5	93.8	100.6	42488.7	43054.5	139.2	136.6	-12.61
30-2- $Psplit$	44367.5	44834.4	78.8	75.9	43463.0	43720.8	59.8	64.0	-2.48
40-2- $Gsplit$	65862.2	67102.7	142.2	153.0	59969.1	60883.6	214.6	236.5	-9.27
40-2- $Psplit$	62346.0	62837.3	101.6	109.1	61598.4	62074.4	67.4	64.6	-1.21
50-2- $Gsplit$	88933.0	90066.5	300.6	246.1	80255.2	82188.4	302.2	321.7	-8.75
50-2- $Psplit$	83313.4	84194.9	209.6	235.1	82294.5	83096.4	58.2	57.4	-1.30
15-3- $Gsplit$	28676.1	29537.9	39.8	43.2	24943.2	25170.0	62.2	76.7	-14.79
15-3- $Psplit$	27886.5	27604.2	33.6	31.9	25528.3	25988.3	54.4	57.6	-5.85
20-3- $Gsplit$	38184.6	39059.2	118.4	117.6	34141.4	34544.8	142.2	145.6	-11.56
20-3- $Psplit$	36238.8	37004.1	73.8	67.60	34538.0	35223.1	53.2	56.8	-4.81
30-3- $Gsplit$	58421.1	59424.7	178.4	178.2	50179	51835.5	185	203.2	-12.77
30-3- $Psplit$	65999.4	67792.9	146.2	144.1	64603.8	65289.1	61.6	73.0	-3.69
40-3- $Gsplit$	88046.5	89404.9	213.4	211.7	73824	75856.5	376.4	368.6	-15.15
40-3- $Psplit$	101664.5	104640.6	387.6	415.8	100183.6	100183.5	67.0	80.6	-4.26
50-3- $Gsplit$	132402.6	134806.4	550.2	454.5	110355.2	114164.3	392.0	368.9	-15.31
50-3- $Psplit$	140470.4	142241.6	569.2	539.6	131556.7	131762.2	83.4	76.1	-7.37
average			184.5	178.2			139.0	141.9	-7.7

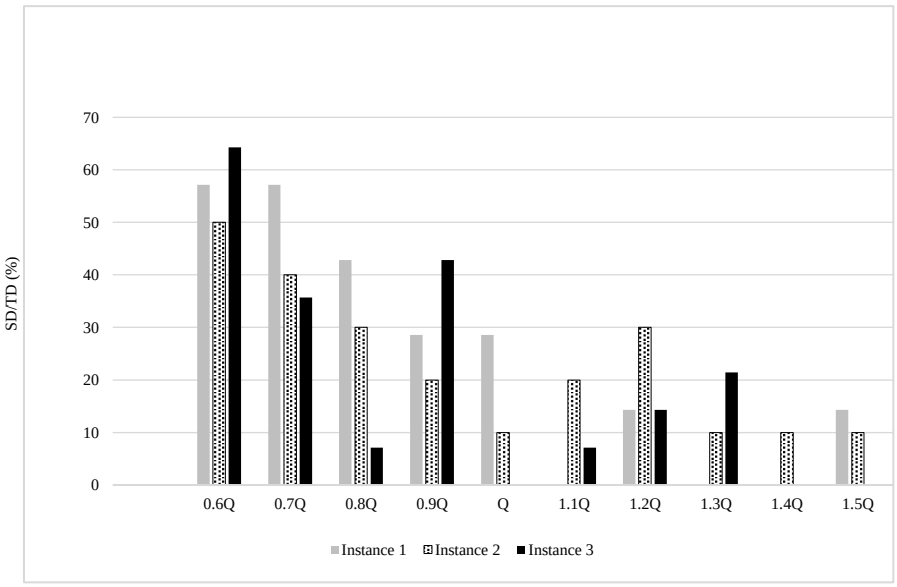

Fig.2. The effect of vehicle capacity on splitting delivery in MCVRP with product-oriented split delivery

In Fig.1, the number of customers (n) is shown along the x-axis, whereas the vertical axis corresponds to the differences between MCVRP with product-oriented split delivery ($Psplit$ mode) and general MCVRP ($Gsplit$ mode). The differences are explained in terms of three factors: the number of vehicles, the travelled distance and the ratio of split demands to total splitable demands (SD/TD). Fig. 1 illustrates how solutions are affected when product-oriented split delivery mode is taken into account. As expected, the average distance travelled and the number of vehicles required in MCVRP with product-oriented split delivery increase compared to the general MCVRP. On average, the number of vehicles is more affected by the mode of demand splitting than distance travelled. The SD/TD in Fig.1 shows that the demand splitting decreases with the increase in the number of customers. In other words, as the number of customers increases, the proposed model approaches the MCVRP without split delivery.

Fig. 2 shows the effect of vehicle capacity on the splitting delivery in three instances (Instance 1: $n = 10$ and $\mathcal{M} = 2$, Instance 2: $n = 15$ and $\mathcal{M} = 2$, Instance 3: $n = 10$ and $\mathcal{M} = 3$). The SD/TD is shown along the y-axis, whereas the horizontal axis corresponds to the capacity of vehicles. Obviously, as the capacity of vehicles increases, the number of vehicles decreases or does not change. At first, the capacity of vehicles is very small compared to the customers' demand, therefore a large part of the orders is delivered in split delivery mode. As vehicle capacity increases, each vehicle can fully serve more customers, thus reducing split deliveries. By further increasing the capacity of vehicles, although each vehicle can visit more customers, it does not fully meet all the demands and again splitting delivery mode increases. Therefore, as the capacity of vehicles increases, a decreasing and then increasing trend in the SD/TD is repeated.

After analyzing the effect of the vehicle capacity on the delivery mode, the influence of the maximum number of compartments ($\mathcal{L}_k$ for $k \in K$), which determines the number of product types per vehicle, is examined in three instances. Table 5 summarizes the effect of decreasing $\mathcal{L}_k$ on the number of vehicles, travelled distance and splitting delivery mode (SD/TD).

Each SCV ($\mathcal{L}_k = 1\ for\ k \in K$) carries only one type of product, while a MCV ($\mathcal{L}_k > 1\ for\ k \in K$) enables joint delivery of several product types. Therefore, if MCVs are considered, customers' demands can be met with less (or equal) number of vehicles. As stated in instance 3, if SCVs are used instead of 3-compartment vehicles, the number of vehicles required will be 14.29% more. As expected, restricting the value of $\mathcal{L}_k$ leads to increased travelled distance. As stated in instance 3, reducing the number of compartments from 3 to 2 leads to 24.8% increase in the travelled distance, while a further reduction from 3 to 1 lead to a further distance increase of 73.43%.

The last column of Table 5 shows the SD/TD in each instance. In MCVs, the vehicle capacity is divided into several compartments; each compartment is assigned to one type of product, while the total capacity of each SCV is assigned to one type of product. As shown in Table 5, split delivery in MCVs is more than SCVs in instance 1. However, it is observed that in instance 3, split delivery in SCVs is more than MCVs. Therefore, it can be conducted that there is no significant relationship between the number of compartments and demand splitting. In addition to the number of compartments and the amount of each customer demand, the number of customers who are visited in each tour also affect the delivery mode.

Table 5
The influence of the number of compartments on the delivery mode

Instance	n	$\mathcal{M}$	$\mathcal{L}_k$	Changes in the number of vehicles compared to $\mathcal{L}_k = \mathcal{M}$ (%)	Changes in the travelled distance compared to $\mathcal{L}_k = \mathcal{M}$ (%)	SD/TD (%)
1	10	2	2			28.57
			1	20	41.97	14.29
2	15	2	2			10
			1	0	32.11	10
3	10	3	3			0
			2	0	24.80	0
			1	14.29	73.43	7.14

6. Conclusion and future research

This paper addressed an MCVRP with split delivery in which the possibility of splitting delivery depends on the product type. Each customer's demand for certain types of products cannot be split, but split delivery of other types of products is permitted if the time interval between vehicles arrival times does not exceed a certain limit. We developed a MIP model that can used to solve small sized instances optimality with the Gurobi optimizer. In order to tackle larger instances, a matheuristic based on fixing a part of customer-to-route assignment variables and an ALNS was proposed.

The proposed algorithms are employed to solve MCVRP with product-oriented split delivery and general MCVRP in which split delivery is allowed for all product types. Computational experiments indicated that in most small sized instances, Gurobi found the same optimal solution in both splitting delivery mode ($Gsplit/Psplit$), indicating that customers' requirements in term of splitting delivery can be implemented without additional costs. Although the delivery mode has an effect on the Gurobi computational time, it does not affect the computational time of the proposed algorithms.

As the problem size increase, the computational time of Gurobi increases exponentially. Both algorithms find near optimal solutions within very short computational times. In the large sized instances, the proposed matheuristic performs better than ALNS with an average improvement of 7.7% and reducing up to nearly 20% of computational time on average. In large sized instances, the product-oriented split delivery in MCVRP may lead to an increase in the number of vehicles needed or affect routing decision and travelled distances. Therefore, the route generated in product-oriented mode are costly than general mode.

Our sensitivity analysis revealed that, on average, the number of vehicles is more affected by delivery mode than distance travelled. Furthermore, the computational results showed that as the number of customer increases, the split delivery decreases and the proposed model approaches the MCVRP without split delivery. However, vehicle capacity, the number of compartments and number of customers in each tour are factors affecting the delivery mode.

An interesting avenue for future research is to consider a heterogeneous fleet composed of various vehicles with different characteristics and costs. In addition, split delivery could be investigated in a fleet of electric or hybrid vehicles. Another aspect for future research could be considering uncertainty in certain parameters (e.g., demand and travel time), which brings the problem closer to real-life conditions. In terms of arrival time consistency, future work could assess the implications of varying the consistency threshold. From the solution approaches perspective, the proposed matheuristic could be developed in future studies.

Compliance with Ethical Standards

Funding details. This research received no specific grants or other support from any organization.

Disclosure statement. The authors report there are no competing interest to declare.

Data availability statement. All used data will be available upon request.

References

Alinaghian, M., & Shokouhi, N. (2018). Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, *76*, 85–99.

Archetti, C., Savelsbergh, M. W. P., & Speranza, M. G. (2008). To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review*, *44*(1), 114–123.

Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, *19*(1–2), 3–22.

Asawarungsaengkul, K., Rattanamanee, T., & Wuttipornpun, T. (2013). A Multi-Size Compartment Vehicle Routing Problem for Multi-Product Distribution: Models and Solution Procedures. *International Journal of Artificial Intelligence*, *11*(A13), 237–256.

Bortfeldt, A., & Yi, J. (2020). The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints. *European Journal of Operational Research*, *282*(2), 545–558.

Cattaruzza, D., Absi, N., Feillet, D., & González-Feliu, J.-J. (2017). Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, *6*(1), 51–79.

Chen, J., & Shi, J. (2019). A multi-compartment vehicle routing problem with time windows for urban distribution – A comparison study on particle swarm optimization algorithms. *Computers & Industrial Engineering*, *133*, 95–106.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91.

Doan, T. T., Bostel, N., & Hà, M. H. (2021). The vehicle routing problem with relaxed priority rules. *EURO Journal on Transportation and Logistics*, *10*, 100039.

Dror, M., & Trudeau, P. (1989). Savings by Split Delivery Routing. *Transportation Science*, *23*(2), 141–145.

Dror, M., & Trudeau, P. (1990). Split Delivery Routing. *Naval Research Logistics*, *37*(3), 383–402.

Dumez, D., Tilk, C., Irnich, S., Lehuédé, F., Olkis, K., & Péton, O. (2023). A Matheuristic for a 2-Echelon Vehicle Routing Problem with Capacitated Satellites and Reverse Flows. *European Journal of Operational Research*, *305*(1), 64–84.

Elatar, S., Abouelmehdi, K., & Essaid, M. (2023). The vehicle routing problem in the last decade: variants , taxonomy The vehicle routing problem in the last decade: variants , taxonomy and metaheuristics. *Procedia Computer Science*, *220*, 398–404.

Friedrich, C., & Elbert, R. (2022). Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics. *Computers and Operations Research*, *137*, 105491.

Goli, A., Aazami, A., & Jabbarzadeh, A. (2018). Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions. *International Journal of Artificial Intelligence*, *16*(1), 88-112.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., & Speranza, M. G. (2024). Vehicle routing problems with multiple commodities: A survey. *European Journal of Operational Research*, *317*(1), 1–15.

Gulczynski, D., Golden, B., & Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, *46*(5), 612–626.

Guo, F., Huang, Z., & Huang, W. (2021). Heuristic approaches for a vehicle routing problem with an incompatible loading constraint and splitting deliveries by order. *Computers and Operations Research*, *134*, 105379.

Han, A. F., & Chu, Y. (2016). A multi-start heuristic approach for the split-delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, *88*, 11–31.

Henke, T. (2018). *Multi-compartment vehicle routing problems in the context of glass waste collection*. Ph.D.

thesis. Magdeburg, Germany: Otto-von-Guericke univresity Magdeburg.

Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research*, *22*(3), 2033–2062.

Kou, S., Golden, B., & Bertazzi, L. (2024). Estimating optimal split delivery vehicle routing problem solution values. *Computers & Operation Research*, *168*, 106714.

Lia, J., Qina, H., Baldaccib, R., & Zhuc, W. (2020). Branch-and-price-and-cut for the synchronized vehicle routingproblem with split delivery, proportional service time and multiple time windows. *Transportation Research Part E: Logistics and Transportation Review*, *140*, 101955.

Mancini, S. (2016). A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Matheuristic. *Transportation Research Part C: Emerging Technologies*, *70*, 100–112.

Martins, S., Ostermeier, M., Amorim, P., Hübner, A., & Almada-Lobo, B. (2019). Product-oriented time window assignment for a multi-compartment vehicle routing problem. *European Journal of Operational Research*, *276*(3), 893–909.

Mohammadi, M., Rahmanifar, G., Hajiaghaei-keshteli, M., Fusco, G., Colombaroni, C., & Sherafat, A. (2023). A dynamic approach for the multi-compartment vehicle routing problem in waste management. *Renewable and Sustainable Energy Reviews*, *184*, 113526.

Moshref-Javadi, M., & Lee, S. (2016). The Customer-centric, multi-commodity vehicle routing problem with split delivery. *Expert Systems With Applications Received*, *56*, 335–348.

Ngoo, C. M., Goh, S. L., Sze, S. N., Sabar, N. R., Hijazi, M. H. A., & Kendall, G. (2024). A survey of mat-heuristics for combinatorial optimisation problems: Variants, trends and opportunities. *Applied Soft Computing*, *164*, 111947.

Ostermeier, M., Henke, T., Hübner, A., & Wäscher, G. (2021). Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research*, *292*(3), 799–817.

Polat, O., & Topaloğlu, D. (2020). Collection of diferent types of milk with multi tank tankers under uncertainty: a real case study. *Top*, *30*(1), 1–33.

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. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. *International Confference on Principles and Practice of Constraint Programming*, 417–431.Berlin, Heidelberg: Springer Berlin Heidelberg.

Toth, P., & Vigo, D. (2014). *Vehicle routing problems: methods and applications* (2nd ed.). Society for Industrial and Applied Mathematics.

Urli, T., & Kilby, P. (2017). Constraint-based fleet design optimisation for multi-compartment split-delivery rich vehicle routing. *Principles and Practice of Constraint Programming: 23rd International Conference, CP 2017, Melbourne,VIC, Australia, August 28-September 1, 2017, Proceedings 23*, 414–430.

Voigt, S. (2024). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research*.

Wang, J., Lian, Z., Liu, C., & Liu, K. (2023). Iterated clustering optimization of the split-delivery vehicle routing problem considering passenger walking distance. *Transportation Research Interdisciplinary Perspectives*, *17*, 100751.

Wang, L., Kinable, J., & van Woensel, T. (2020). The fuel replenishment problem: A split-delivery multi-compartment vehicle routing problem with multiple trips. *Computers & Operations Research*, *118*, 104904.

Zbib, H., & Laporte, G. (2020). The commodity-split multi-compartment capacitated arc routing problem. *Computers and Operations Research*, *122*, 104994.

Zhang, Z., Che, Y., & Liang, Z. (2024). Split-demand multi trip vehicle routing problem with simultaneous pickup and delivery in airport baggage transit. *European Journal of Operational Research*, *312*(3), 996–1010.

Appendix: Detailed results of small sized instances

Table A1
The detailed results of small sized instances

		Gurobi			ALNS					Matheuristic					
EX	Instance	Obj^*	Gap^* (%)	T (s)	Obj_{min}	$\overline{Obj}$	$\overline{Gap}$ (%)	T_{min} (s)	$\bar{T}$ (s)	Obj_{min}	$\overline{Obj}$	$\overline{Gap}$ (%)	T_{min} (s)	$\bar{T}$ (s)	$\overline{Imp}$ (%)
1	10-2- $Gsplit$	7977.7	0.00	4.7	7977.7	7977.7	0.00	3	3.6	7977.7	7977.7	0.00	10.0	10.4	0.00
	10-2- $Psplit$	7977.7	0.00	6.7	7977.7	7977.7	0.00	4	4.4	7977.7	7977.7	0.00	6.0	8.9	0.00
2	10-2- $Gsplit$	10562.5	0.00	7.4	10562.5	10562.5	0.00	5	7	10562.5	10562.5	0.00	13.0	17.4	0.00
	10-2- $Psplit$	10562.5	0.00	6.4	10562.5	10562.5	0.00	4.7	4.7	10562.5	10562.5	0.00	8.4	10	0.00
3	10-2- $Gsplit$	15386.9	0.00	516.1	15386.9	15546.9	1.04	10	10.4	15386.9	15386.9	0.00	23.0	25.4	-1.03
	10-2- $Psplit$	15386.9	0.00	117.6	15386.9	15386.9	0.00	5.2	7.1	15386.9	15386.9	0.00	10.0	4.4	0.00
4	10-2- $Gsplit$	14146.4	0.00	33.8	14146.4	14188.7	0.30	10	8.6	14146.4	14146.4	0.00	20.0	21.0	-0.30
	10-2- $Psplit$	14146.4	0.00	24.8	14146.4	14146.4	0.00	5.2	6.3	14146.4	14146.4	0.00	10.0	12.0	0.00
5	10-2- $Gsplit$	14368.5	0.00	129.7	14368.5	14487.6	0.83	9	7	14368.5	14370.2	0.01	23.0	27.8	-0.81
	10-2- $Psplit$	14368.5	0.00	115.3	14368.5	14368.5	0.00	4.6	5.9	14376.9	14376.9	0.06	10.0	10	0.06
1	10-3- $Gsplit$	8105.2	0.00	1.8	8105.2	8105.2	0.00	6	7	8105.2	8105.2	0.00	9.0	9.4	0.00
	10-3- $Psplit$	8105.2	0.00	3.5	8105.2	8105.2	0.00	4.2	5.2	8105.2	8105.2	0.00	4.0	5.6	0.00
2	10-3- $Gsplit$	16403.4	0.00	296.4	16526.2	16542.4	0.85	11	10.8	16403.4	16403.4	0.00	23.0	30.0	-0.84
	10-3- $Psplit$	16403.4	0.00	621.6	16403.4	16419.6	0.10	5.3	7.9	16403.4	16403.4	0.00	20.0	26.0	-0.10
3	10-3- $Gsplit$	18036.9	0.00	536.4	18036.9	18206.9	0.94	12	15	18036.9	18036.9	0.00	30.0	38.0	-0.93
	10-3- $Psplit$	18036.9	0.00	557.5	18036.9	18186.0	0.83	6.5	6.8	18036.9	18126.6	0.50	20.0	24.8	-0.33
4	10-3- $Gsplit$	17377.9	0.00	458.8	17541.1	17443.2	0.38	13	11.2	17377.9	17377.9	0.00	23.0	28.2	-0.37
	10-3- $Psplit$	17377.9	0.00	2131.9	17377.9	17391.3	0.08	6.3	6.5	17377.9	17386.9	0.05	20.0	32.4	-0.03
5	10-3- $Gsplit$	16059.0	0.00	890.1	16059.0	16144.8	0.53	18	16	16059.0	16165.9	0.67	21.0	24.2	0.13
	10-3- $Psplit$	16059.0	0.00	430.8	16059.0	16217.4	0.99	6	7.7	16059.0	16059.0	0.00	20.0	20.0	-0.98
1	15-2- $Gsplit$	17892.7	0.00	12764.1	17892.7	17996.9	0.58	14	12.7	17892.7	18001.0	0.61	30.0	35.8	0.02
	15-2- $Psplit$	18413.5	0.00	11617.9	18413.5	18434.9	0.12	7.5	8.4	18413.5	18413.5	0.00	21.0	23.0	-0.12
2	15-2- $Gsplit$	20675.1	0.00	5096.8	20675.1	20878.6	0.98	13.8	15.7	20675.1	20697.1	0.11	67.0	68.2	-0.87
	15-2- $Psplit$	20682.3	0.00	2289.8	20979.5	20979.5	1.44	8.2	12.3	20773.1	20850.7	0.81	31.0	37.2	-0.61
3	15-2- $Gsplit$	16899.5	0.00	3229.0	16899.5	17044.5	0.86	12.8	15.4	16899.5	16914.1	0.09	31.0	32.4	-0.77
	15-2- $Psplit$	16935.9	0.00	808.2	16999.1	17130.3	1.15	12	13.3	16935.9	16958.5	0.13	34.0	42.2	-1.00
4	15-2- $Gsplit$	18691.5	0.00	23814.4	18691.5	18843.5	0.81	14	15.3	18691.5	18696.2	0.03	31.0	36.4	-0.78
	15-2- $Psplit$	18691.5	0.00	9796.3	18691.5	18818.4	0.68	10	10.9	18691.5	18710.0	0.10	20.0	27.2	-0.58
5	15-2- $Gsplit$	18264.3	0.00	62.2	18264.3	18391.3	0.70	13	12.6	18264.3	18264.3	0.00	32.0	50.2	-0.69
	15-2- $Psplit$	18264.3	0.00	37.3	18264.3	18391.3	0.70	11	11.7	18264.3	18322.5	0.32	31.0	29.8	-0.37
average			0.00	2546.9			0.50	8.8	9.6			0.12	21.7	25.6	-0.38