European Journal of Operational Research 324 (2025) 477–491

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

Last mile delivery routing problem with some-day option

Stefan Voigt *, Markus Frank, Heinrich Kuhn

Catholic University of Eichstätt-Ingolstadt, Ingolstadt School of Management, Auf der Schanz 49, 85049 Ingolstadt, Germany

ARTICLE INFO

Keywords:
Routing
E-commerce
Last mile delivery
Vehicle routing problem
Adaptive large neighborhood search

ABSTRACT

E-commerce retailers are challenged to maintain cost-efficiency and customer satisfaction while pursuing sustainability, especially in the last mile. In response, retailers are offering a range of delivery speeds, including same-day and instant options. Faster deliveries, while trending, often increase costs and emissions due to limited planning time and reduced consolidation opportunities in the last mile. In contrast, this paper proposes the inclusion of a slower delivery option, termed some-day. Slowing down the delivery process allows for greater shipment consolidation, achieving cost savings and environmental goals simultaneously. We introduce the dynamic and stochastic some-day delivery problem, which accounts for a latest delivery day, customer time windows, and capacity limitations within a multi-period planning framework. Our solution approach is based on addressing auxiliary prize-collecting vehicle routing problems with time windows (PCVRPTW) on a daily basis, where the prize reflects the benefit of promptly serving the customer. We develop a hybrid adaptive large neighborhood search with granular insertion operators, outperforming existing metaheuristics for PCVRPTWs. Our numerical study shows significant cost savings with only small increases in delivery times compared to an earliest policy.

1. Introduction

Parcel deliveries are at an all-time high, amounting for 161 billion parcels worldwide in 2022 (Pitney Bowes, 2023), driven mainly by the continuous growth of e-commerce. This surge in online shopping is accompanied by a growing demand for even faster deliveries in the business-to-consumer (B2C) sector. Although the standard delivery time from retailer to the customer's home typically falls within the range of one to three days for most deliveries, the segments experiencing the fastest growth are same-day and instant delivery (Deloison et al., 2020). For instance, as early as 2019, Amazon managed to deliver 72% of its U.S. customers within 24 h (Kim, 2019). However, short delivery times put tremendous pressure on traditional transportation networks and often lead to less efficient distribution processes. This inefficiency is characterized by poorly and unevenly utilized resources, greatly increased costs and emissions per parcel. In particular, this is caused by a lack of consolidation possibilities and insufficient time for planning. To address this challenge, for instance Dumez et al. (2021) introduce delivery location options to foster consolidation.

In this paper, we explore a straightforward and potentially rapid implementation of a delivery concept for B2C parcel delivery. The primary goal is to reduce delivery costs and thus the environmental impact. This can be achieved by deliberately slowing down the logistics processes involved in parcel delivery, thereby allowing for the consolidation of more shipments over an extended time period. The idea originates from the concept of *slow logistics*. Slow logistics covers various methods and approaches associated to logistical activities in supply chains which explicitly make use of available time potentials, tolerating a slowdown of processes in order to reduce costs and adverse environmental impacts (Wiese, 2017). A well-known example of a slow logistics instrument is slow steaming, where ocean container carriers intentionally reduce vessel speeds to enhance fuel efficiency and reduce greenhouse gas emissions (see, e.g., Maloni et al., 2013). The practice of consolidating shipments is another commonly employed technique that plays a major role in nearly all applications of slow logistics concepts. Nevertheless, it is important to note that slow logistics concepts have primarily found application in supply chain problem scenarios within business-to-business industries.

Applications in the last-mile delivery of parcels in the B2C sector have been relatively rare, but notable exceptions exist. For instance, Amazon already provides a "free no-rush" delivery option in many U.S. regions (Amazon, 2022). Customers who select this option accept a longer delivery time frame and, in return, receive a discount on their order or another type of reward. Alternatively, a significant portion of customers may be encouraged to accept the potential delay by highlighting the reduced emissions associated with slower delivery (Dietl et al., 2024).

* Corresponding author.
E-mail addresses: stefan.voigt@ku.de (S. Voigt), markus.frank@ku.de (M. Frank), heinrich.kuhn@ku.de (H. Kuhn).

https://doi.org/10.1016/j.ejor.2025.02.001
Received 11 May 2024; Accepted 1 February 2025
Available online 11 February 2025

In the following, we assume the point of view of an e-commerce retailer who operates its own delivery fleet and is thus able to directly benefit from greater flexibility in delivery dates. The retailer offers a range of delivery options to their customers, including a notably slower *some-day* alternative. Each customer's order is subject to specific delivery date constraints, comprising both an earliest and a latest possible delivery day that must be strictly adhered to.

We introduce the *some-day delivery problem* (SDDP) to assess the implications of implementing a some-day delivery option for tour planning. The SDDP aims to determine the most cost-effective delivery day for each customer, as well as the clustering and routing. Our modeling assumptions align with the broader category of multiple period VRPs (MPVRP, see Archetti et al., 2015). In practice, customer orders arrive continuously throughout the planning period, creating a dynamic setting. While a similar dynamic MPVRP, incorporating constraints on delivery days, has been proposed by Wen et al. (2010), our situation differs in that not all relevant information is known in advance but may be available in a stochastic manner such as forecasts on customer demands and their geographical distribution. The goal is to leverage these available data when assigning delivery days to customers. Consequently, the SDDP falls within the domain of dynamic and stochastic vehicle routing problems with multiple periods.

The paper contributes to the current literature by (1) describing a novel slow logistics concept for B2C parcel delivery, (2) reviewing and categorizing existing work in the field of MPVRPs with delivery dates, (3) introducing a solution approach for a dynamic MPVRP that utilizes stochastic information, (4) implementing a powerful hybrid adaptive large neighborhood search with granular insertion operators for solving an auxiliary prize-collecting VRPTW, and (5) show by simulation that a slow delivery option significantly improves costs.

The remainder of this paper is organized as follows: Section 2 describes the dynamic and stochastic some-day delivery problem. In Section 3, we offer an overview of related literature. We describe a static and deterministic model in Section 4 and a solution approach for the dynamic and stochastic extension in Section 5. In Section 6, we conduct extensive numerical experiments. Finally, in Section 7, we summarize our findings and conclude our work.

2. Some-day delivery problem: Concept, costs and constraints

2.1. Last mile delivery concept with slow delivery option

We propose to slow down the delivery process to enable a more efficient customer clustering in the time dimension. We achieve this by introducing a significantly slower delivery option for customers, which we refer to as the some-day delivery option. This some-day option is presented during the delivery method selection at the online checkout and ensures that customers receive their orders by a specified latest delivery day. However, the exact delivery day is only revealed to the customer once their order has been finalized for delivery. Importantly, we do not compromise customer satisfaction by altering their freedom to choose their preferred delivery method. Instead, the retailer may offer this some-day option as an additional choice alongside existing ones.

From a last mile perspective, the some-day delivery option increases flexibility for a retailer operating its own vehicle fleet. When customers select the some-day delivery option, their orders can be delivered on any day within the specified delivery interval. Within this time interval, the retailer can allocate orders to the best (i.e., cost-effective and eco-friendly) delivery day, respecting orders with fixed delivery dates (e.g., next-day orders) and other logistical constraints. However, the some-day delivery option could potentially lead to an increase in failed delivery attempts, as it does not allow customers precise control over the exact day of delivery. Therefore, the benefits of this option must outweigh any potential rise in the costs associated with failed deliveries. Nevertheless, the retailer may have some knowledge of the availability profile of flexible customers and could take this into account when setting up its delivery schedule (Voigt et al., 2023).

Fig. 1 provides three illustrative scenarios. In cases where fixed delivery dates are imposed, tours on two days are scheduled less efficiently in terms of costs (case 1) or uneven capacity utilization (case 2). In contrast, the scenario including customers who selected the some-day delivery option, demonstrates how this added flexibility can reduce costs and emissions associated with inefficient tours. Moreover, it allows for a more even distribution of demands across multiple days. Obviously, the flexibility effects grow with the share of customers selecting the some-day delivery option. However, it is important to note that as the retailer now has to make decisions about the delivery days for these orders, the complexity of tour planning rises with the number of days. Thus, a decision problem emerges where the retailer has to decide (1) on the delivery day for each flexible order, (2) on the clustering of orders into tours on each day, and (3) on the sequence on those tours.

The problem setting described is not exclusive to e-commerce retailers with their own delivery fleets but also extends to parcel service providers, given they have the capability to temporarily store shipments and access information about the latest delivery dates. Parcel service providers can gather these information through various means, including offering differently prioritized delivery options. For example, DHL provides a slower delivery option with a transit time of four or more days, which is well-suited for shipping e-commerce goods (DHL, 2021). Alternatively, the parcel service provider could collaborate with an e-commerce retailer to obtain the requested delivery intervals.

2.2. Consumer behavior

One fundamental assumption in our problem setting is that customers would opt for a slower some-day delivery option, even though literature suggests that e-commerce customers typically prefer receiving their orders as quickly as possible (e.g., Nogueira et al., 2021). To encourage customers to choose the some-day option, retailers can employ several strategies, such as offering monetary incentives like lower shipping costs, discounts, or other rewards (see, e.g., Amazon, 2022). Another effective approach that has yet to be fully utilized is to appeal to customers' environmental consciousness by providing information about the ecological benefits of opting for slower delivery at the checkout. Recent studies indicate that a majority of e-commerce customers is open to choosing a slower delivery option. For instance, Buldeo Rai et al. (2019) found that more than two-thirds of customers may be willing to wait longer for their parcels if it leads to fewer kilometers driven or if all parcels from a single order transaction are delivered together. However, it is essential to explicitly display the environmental impacts of each delivery option, as customers are often not fully aware of them. Ignat and Chankov (2020) demonstrated that the probability of an online customer choosing a slower delivery increased significantly, from 4% to 66%, after being provided with information on its environmental impact. Furthermore, Buldeo Rai et al. (2021) discovered that the percentage of customers selecting the slower delivery option could be further boosted through additional measures, such as indicating that many others have already chosen this option. While the findings of Nogueira et al. (2021) confirm the willingness of e-commerce customers to wait, they also reveal differences based on the type of product ordered. Additionally, Dietl et al. (2024) found that the effectiveness of incentives on the willingness to wait depends on individual characteristics and socio-demographics. In particular, customers with a strong environmental awareness are more willing to wait longer when presented with emission savings associated with delayed delivery. Another potential incentive to increase the willingness to wait is to offer customers greater flexibility in their delivery choices, including the option to select preferred delivery time windows (TWs).

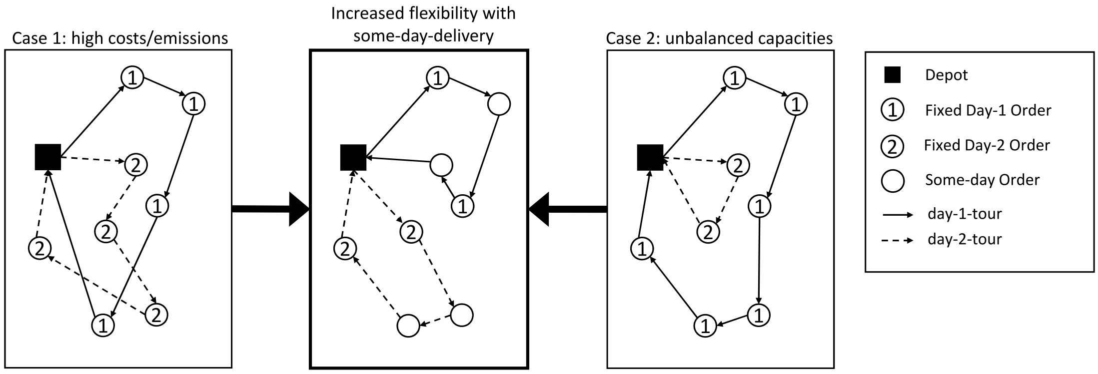


Fig. 1. Flexibility effects through some-day delivery option.

2.3. Decision-relevant costs and constraints

In order to formalize the decision problem stemming from the last mile delivery concept with slow delivery option outlined above, we examine decision-relevant costs and constraints in the following.

Multi-period planning horizon. The planning horizon is divided into multiple periods, each representing a day. For clarity, we will use *days* as the term for each delivery period to distinguish it clearly from time windows (discussed in the next paragraph). It is important to note that the planning horizon is considered infinite.

Demand, time window, and delivery interval for each order. Customers continuously place their orders throughout the planning horizon, specifying their demand, eligible delivery days by selecting from available delivery options, and potentially a delivery time window (TW) during which delivery is possible on the specified day. Each delivery option includes a delivery interval, which consists of the earliest possible and latest allowable delivery days. The earliest is the day following the order placement, excluding same-day and instant delivery options. For example, if a customer selects a standard option promising delivery within one to three days, delivery can occur as early as day one and not later than day three, but the customer cannot exert more precise control over the specific delivery day. Even customers who intentionally opt for slower delivery options may prefer a faster delivery over a slower one. To account for this, calculative penalty costs, referred to as **waiting costs**, may be applied, occurring per order and per day.

Delivery structures. We assume a single distribution center (DC) with sufficient capacity where orders are picked and packed as soon as they arrive. This setup allows to offer a next-day delivery option. However, orders which are not scheduled for the following day must be temporarily stored in designated areas at the DC (possibly over multiple days) until they are scheduled for delivery. Therefore, **inventory costs** that occur per order and per day may be taken into account.

Vehicle fleet. The retailer operates its own vehicle fleet for last mile delivery and is therefore able to flexibly (re-)arrange orders to delivery tours on any day. Each vehicle has a capacity that may not be exceeded and must arrive at the DC within a time limit. **Transportation costs** arise for conducting the delivery tours, depending on the distances between the customer locations. These costs include factors such as fuel consumption, driver wages, and also vehicle emissions. The number of vehicles to be used each day is restricted by the fleet's size. Additionally, there is a contingency plan in place, such as an external logistics provider, to prevent the violation of latest delivery days. This backup option becomes particularly relevant in scenarios with stochastic customer orders and high utilization of vehicles. **Backup costs** are incurred for each order where the backup option must be employed. These backup costs are independent of each other, similar to penalty costs for unserved orders in Archetti et al. (2015).

2.4. Order information quality and availability

Customer order data play a central role in the problem setting. Four main characteristics of an order can be identified: the customer's location, demand, delivery TW, and the selected delivery option (i.e., delivery days interval). These characteristics, along with the overall set of orders, determine the feasible delivery tours on each day. However, these details are revealed only as orders are placed, which occurs gradually over time. Consequently, the order data are not known with certainty for the entire planning horizon when delivery tours are being planned. However, we assume, that the retailer has stochastic information on the **expected number of orders on future days** as well as on the **geographical demand distribution** (i.e., customer density and related probabilities on order frequencies). In contrast, we assume that customer individual demands, delivery TWs and selected delivery options are highly stochastic and are only revealed when the customer places the order. As such, these information cannot be utilized for planning.

2.5. Summary

We adopt the perspective of an e-commerce retailer with its own delivery fleet or a collaborating parcel service provider that receives information about requested (latest) delivery days. Customers can select from various delivery options when placing their orders, including a considerably slower alternative, termed some-day option. Customers may opt for this slower option due to monetary incentives, sustainability concerns, or due to additional incentives like the option to select delivery TWs. This some-day option provides the retailer or parcel service provider with increased flexibility regarding when to make the delivery, thus offering more opportunities to consolidate shipments from the customer's vicinity over time. We assume that orders arrive gradually over time but probabilistic information is available about the number of future customer orders and their geographical distribution. In such a dynamic setting, the ability to anticipate future orders becomes crucial in finding feasible tour plans, avoiding penalties for exceeding capacities, and capitalizing on consolidation opportunities. The decision model for this problem setting determines the delivery day for each order and plans the clustering and routing of orders with a constrained vehicle fleet across multiple days. The objective is to devise a tour plan that minimizes costs, encompassing waiting costs/inventory costs, transportation costs and penalty costs for backup deliveries. Waiting and inventory costs can be combined into a single cost parameter, as they depend on the same decision. The trade-off revolves around balancing waiting and inventory costs, which favor delivering as soon as possible, against transportation and backup costs, which tend to favor slower, consolidated deliveries. We henceforth refer to the setting described as the some-day delivery problem (SDDP).

3. Literature

Conceptually, the problem is related to vehicle routing problems with delivery options (e.g., Dragomir et al., 2022; Dumez et al., 2021; Tilk et al., 2021). However, this stream of literature focuses on enhancing flexibility during the day by introducing options for the delivery location. In contrast, the SDDP enhances flexibility by allowing the service provider to freely select the delivery day within an earliest and latest delivery day. As such, the problem setting belongs to the class of multi-period vehicle routing problems (MPVRP). The classical MPVRP considers a planning period consisting of several days, where each customer requires a single visit and a fixed number of capacitated vehicles is available each day to fulfill these customer demands (e.g., Archetti et al., 2015). As there exists an abundance of literature on problems with multiple periods (see Campbell & Wilson, 2014; Francis et al., 2008), we focus the following review on publications related to MPVRPs that closely align with our problem setting. We deem a publication as related if it restricts customer visits to individual delivery intervals, defined by a release and/or due date. Table 1 presents the reviewed literature, structured according to the taxonomy introduced by Bektaş et al. (2014) in the context of dynamic vehicle routing. Note that settings featuring dynamic and/or stochastic elements are of particular interest, as our problem includes dynamic and stochastic order information.

- *Static and deterministic:* All orders are known in advance with certainty. A comprehensive decision model can be formulated and solved once, providing delivery tour plans valid for the entire planning horizon.
- *Static and stochastic:* At least some information such as demand quantity, selected delivery options, customer locations, or the number of orders placed on any given day is stochastic. However, the probabilities and distributions are known in advance and remain constant throughout the planning horizon. Decision making still occurs only once, but it must account for the stochasticity through recourse actions. Appropriate approaches are chance-constrained models or robust optimization techniques.
- *Dynamic and deterministic:* Information is not fully available when planning begins but is partially revealed over time. Initial tour plans are constructed based on the orders known at the start of the planning horizon. As additional orders arrive on subsequent days, re-planning actions are required, either instantly as new information becomes available or periodically (e.g., daily) using a rolling horizon approach.
- *Dynamic and stochastic:* Re-planning actions are necessary to account for order information that becomes available during the planning horizon. However, probabilistic information about future orders is available, such as weekly seasonality, which can be integrated into the decision-making process to anticipate uncertain orders or their characteristics.

Table 1
Overview of reviewed MPVRPs with delivery intervals.

	Deterministic	Stochastic
Static	Ceschia et al. (2010) Pacheco et al. (2012) Archetti et al. (2015) Mancini (2016) Cantu-Funes et al. (2017) Darvish et al. (2019) Larrain et al. (2019) Estrada-Moreno et al. (2019) Yildiz and Savelsbergh (2020) Muñoz-Villamizar et al. (2021)	Darvish et al. (2020) Subramanyam et al. (2021)
Dynamic	Angelelli, Grazia Speranza et al. (2007) Angelelli, Savelsbergh et al. (2007) Angelelli et al. (2009b) Angelelli et al. (2009a) Angelelli et al. (2010) Wen et al. (2010) Laganà et al. (2021)	Albareda-Sambola et al. (2014) Ulmer et al. (2018) Billing et al. (2018) Ulmer (2018) Keskin et al. (2023)

3.1. Static MPVRP settings

3.1.1. Static and deterministic settings

Most research on MPVRPs with delivery intervals falls within the area of static and deterministic settings. Ceschia et al. (2010) consider an MPVRP arising from a real-world application. They are the first to include delivery intervals (i.e., earliest and latest allowed delivery day) in an MPVRP. Compared to the SDDP, delivery intervals are treated as soft constraints only, and no inventory/waiting costs are assumed. Similarly, the study by Cantu-Funes et al. (2017) incorporates a latest delivery day among numerous other constraints, all driven by a complex real-world application. Mancini (2016) formulates a rich MPVRP with allowed delivery days, motivated by a real-life problem setting, too. On any given day, vehicles have the flexibility to start their routes from one depot and conclude them at another one, provided it optimizes the subsequent day's delivery schedule. Again, only transportation costs are considered. Pacheco et al. (2012) investigate the flexibility of delivering earlier in an MPVRP. Again, additional costs that depend on the selected delivery day are missing. Archetti et al. (2015) introduce an MPVRP with due dates motivated by a city logistics problem. In their model, transportation costs are accompanied by inventory costs incurred for each day an order remains at the depot beyond its release date. The MPVRP with due dates of Archetti et al. (2015) is the model closest to our setting and could be further adapted to fully match the requirements for a static and deterministic variant of the SDDP. Building on the work of Archetti et al. (2015) and Larrain et al. (2019) present two new solution approaches for the same problem setting and tackle instances with up to 100 customers. Darvish et al. (2019) extend the setting of Archetti et al. (2015) by introducing several depots, from which one is selected for delivery. Muñoz-Villamizar et al. (2021) link the MPVRP with due dates to ecological factors. No additional features compared to the setting of Archetti et al. (2015) are considered. Estrada-Moreno et al. (2019) introduce price discounts for delivery flexibility. Customers specify a preferred delivery day, not guaranteed by the service provider. Instead, customers receive a fixed price discount if the delivery occurs on any other day. Yildiz and Savelsbergh (2020) also focus on customer discounts to increase flexibility but determine discounts based on routing rather than prespecified rates. They assume a single vehicle serving customers with delivery locations on a line, simplifying the routing.

3.1.2. Static and stochastic settings

MPVRP models that are stochastic and static at the same time are quite hard to find in literature, let alone such models that also include delivery dates. This scarcity may be attributed to the fact that static-stochastic optimization models typically adhere to the a priori optimization approach, where a stochastic model is solved only once. However, in a multi-period context, there is a high likelihood that some stochastic elements will reveal their actual values over time, allowing for the opportunity to re-design solutions for later periods based on new information. Darvish et al. (2020) present an MPVRP with a single vehicle arising in the context of e-commerce parcel delivery. Orders can generally be delivered in an interval spanning from the day where the order is placed to a specified latest delivery day. However, the availability of products to be delivered at the depot each day follows a given probability distribution. The delivery dates must be communicated to the customers already at the beginning of the planning horizon. This setting can be regarded as a variant of the SDDP, where only the lower bound of the delivery interval (i.e., the earliest possible delivery day) is subject to stochasticity. Subramanyam et al. (2021) study an MPVRP where customers can place their orders on any given day within the planning horizon, thereby revealing demand quantities and allowed delivery days. The authors state that their solution approach would be implemented in a rolling horizon fashion (i.e., dynamically), where order information is updated and delivery tours are re-optimized. In their modeling approach, however, it is assumed that only customers who have already placed their orders are known.

3.2. Dynamic MPVRP settings

3.2.1. Dynamic and deterministic settings

The literature on dynamic and deterministic multi-period settings is rather scarce, even though we were able to identify seven contributions that can be assigned to our categorization (see also Psaraftis et al., 2015). Angelelli, Grazia Speranza et al. (2007) introduce the dynamic and deterministic MPVRP, where orders arrive at the beginning of each day. The planner must decide which orders to fulfill on the current day and which to postpone to the next day, without knowing the set of new requests that will emerge the following day. This scenario involves a single vehicle with unlimited capacity available at a central depot for making deliveries, with the objective of minimizing total transportation costs. The authors present three straightforward algorithms and assess their performance. These are further analyzed in Angelelli, Savelsbergh et al. (2007) for a special case. The uncertainty in this setting corresponds to the uncertainty within the SDDP regarding the number of future customers and their locations. However, unlike the SDDP, they assume that the delivery interval of future customers is known. Angelelli et al. (2009a) expand on their previous works by introducing a set of uncapacitated vehicles and a maximum route length constraint. Moreover, some orders may arrive during the day, allowing tour plans to be modified while vehicles are already en route between customers. Unlike their previous works this extension includes unpostponable same-day orders. Similar to the SDDP approach, as orders are never rejected, some need to be directed to a backup option at a higher cost. This group of authors further extends their prior work by implementing a rolling horizon solution framework (Angelelli et al., 2009b) and conducting additional numerical experiments (Angelelli et al., 2010). Wen et al. (2010) formulate an extended variant of the dynamic MPVRP where orders arrive over time and must be delivered within customer-specific delivery intervals. The objective function considers transportation costs, customer waiting costs, and workload balancing. This setting closely resembles a dynamic-deterministic case of the SDDP, as it includes all relevant constraints and a similar objective function. The primary distinction is the absence of a backup delivery option and that stochastic forecasts on future demand and potential customer locations are not taken into account. Laganà et al. (2021) define a general dynamic MPVRP where both vertices and edges in the mixed graph can serve as delivery destinations. This setting arises in combined postal and parcel delivery when no prior knowledge of future demand is available.

3.2.2. Dynamic and stochastic settings

The field of dynamic and stochastic VRPs has gained significant attention in recent years, as evident in the comprehensive reviews conducted by Ritzinger et al. (2015) and Soeffker et al. (2022). Nevertheless, the combination of multiple periods and delivery intervals, characterizing the SDDP, has not been as frequently explored. Albareda-Sambola et al. (2014) extend the deterministic model of Wen et al. (2010) and add the assumption that for each customer, a probability distribution is known indicating the likelihood of the customer requiring service in future periods. However, they consider demand quantities to be fixed and known in advance. In contrast, we assume the customer set is unknown, but we have access to a probability distribution describing the potential demand. Furthermore, their objective function only includes transportation costs. They propose an adaptive policy for determining which customer to serve each day and which to postpone. This is achieved by solving an auxiliary PCVRP based on a customer compatibility index that reflects the potential savings if two customers are served together and is used to define an appropriate prize. Our solution approach builds on this concept by introducing a method of determining the prize suitable for the SDDP. Billing et al. (2018) present a model motivated by the delivery of vehicles to dealers. The dealers make up a fixed set of known customers that place their requests on a daily basis and without prior notice. However, historical orders from a customer can be used to forecast future ones. Probabilities are provided for each customer to place an order on the next day, but the associated demands and deadlines remain unknown. The objective is to minimize transportation costs consisting of costs for traveled distances and fixed costs per tour and per customer stop. To address this problem, the authors propose a three-phase heuristic based on a customer compatibility index like in Albareda-Sambola et al. (2014). Ulmer et al. (2018) introduce an MPVRP where same-day orders can be postponed by one day similar to the deterministic setting outlined by Angelelli et al. (2009a). Each day, some customers are known in advance and must be served on the same day. Additional customer orders arrive during the day according to a known stochastic distribution. The objective is to minimize the number of postponed orders. In a subsequent work, Ulmer (2018) develops a new solution approach for the model defined in Ulmer et al. (2018). Keskin et al. (2023) consider an MPVRP with due dates and dynamically arriving customer orders. They use predictive information for demand management, including customer calls when early service may be beneficial. This demand management aspect represents a unique feature in their paper.

4. Static and deterministic model formulation

In the following, we formulate a static and deterministic definition of the SDDP. The model is inspired by an MPVRP proposed by Archetti et al. (2015). We intentionally present a more straightforward (but computationally more demanding) flow based formulation, as the goal of this section is to clearly define our problem including the objective function and all constraints. The central idea of model SDDP is therefore the unambiguous definition of the formal problem setting under consideration, even though the stochastic-dynamic environment assumed later is initially neglected.

The problem is defined on a directed graph $G(N, A)$ with node set N and arc set A. Node set N consists of the depot 0 and customers $i \in C$. The arc set is defined as $A = \{(i,j) : i \neq j, i, j \in N\}$. The parameter c_{ij}^{trans} denotes the associated costs with each arc, c_i^{inv} denotes the inventory/waiting costs per day and customer, and c_i^{back} denotes the costs when the customer is served by a backup option. A fleet K of homogeneous vehicles with capacity Q is available to perform routes starting and ending at the depot on each day. The vehicle has to arrive at the depot within the deadline D. A customer's demand d_i can be delivered on any day within its allowed delivery interval $[e_i, l_i]$, where $e_i \leq l_i$ and $l_i \leq T$. On the selected delivery day the service which takes service time S_i must begin within the customer's designated TW $[etw_i, ltw_i]$. For simplicity, we assume that the time window remains consistent across all days. However, this assumption is not essential for the subsequent models or solution approach. If multiple orders from the same customer are considered, this can be modeled by co-locating customers. Table 2 summarizes the notation.

Model static and deterministic SDDP

$$Minimize \quad \sum_{k\in K}\sum_{i,j\in N}\sum_{t\in[e_i,l_i]} c_{ij}^{\text{trans}} \cdot x_{ijkt} + \sum_{i\in C} c_i^{\text{inv}} \cdot \sum_{t\in[e_i+1,l_i]} (t-e_i) \cdot \left(y_{it} + \sum_{k\in K} z_{ikt} \right) + \sum_{i\in C} c_i^{\text{back}} \cdot \sum_{t\in[e_i,l_i]} y_{it} \tag{1}$$

s.t.

$$\sum_{k\in K}\sum_{j\in N}\sum_{t\in[e_i,l_i]} x_{ijkt} + \sum_{t\in[e_i,l_i]} y_{it} = 1 \qquad \forall i \in C \tag{2}$$

Table 2
Notation used to model static and deterministic SDDP.

Sets	
C	Set of customers $C = \{1, \dots, \|C\|\}$
N	Set of nodes $N = \{0\} \cup C = \{0, \dots, \|C\|\}$
K	Set of vehicles $K = \{1, \dots, \|K\|\}$
T	Set of days $T = \{1, \dots, \|T\|\}$
Parameters	
c_{ij}^{trans}	Transportation costs for traveling from location i to j; $i, j \in N$
c_i^{inv}	Inventory/waiting costs for storing the order of customer i for one day; $i \in C$
c_i^{back}	Backup costs when customer i is delivered via backup option; $i \in C$
d_i	Demand size of customer i; $i \in C$
e_i	Earliest delivery day of customer i; $i \in C$
l_i	Latest delivery day of customer i; $i \in C$
τ_{ij}	Time for traveling from location i to j; $i, j \in N$
S_i	Service time at customer i; $i \in C$
etw_i	Earliest delivery time of customer i; $i \in C$
ltw_i	Latest delivery time of customer i; $i \in C$
D	Latest arrival time at depot
Q	Vehicle capacity
Decision and auxiliary variables	
x_{ijkt}	Indicates whether arc (i, j) is traversed by vehicle k on day t; $i, j \in N, k \in K, t \in T$
z_{ikt}	Indicates whether customer i is served by vehicle k on day t; $i \in C, k \in K, t \in T$
y_{it}	Indicates whether customer i is served by a backup option on day t; $i \in C, t \in T$
s_{ikt}	Start time of service of vehicle k at node i on day t; $i \in N, k \in K, t \in T$

$$\sum_{i\in C} d_i \sum_{j\in N} x_{ijkt} \le Q \qquad \forall k \in K, t \in T \tag{3}$$

$$\sum_{j\in N} x_{ijkt} = \sum_{j\in N} x_{jikt} \qquad \forall i \in N, k \in K, t \in T \tag{4}$$

$$z_{ikt} = \sum_{j\in N} x_{ijkt} \qquad \forall i \in C, k \in K, t \in T \tag{5}$$

$$\sum_{j\in N} x_{0jkt} \le 1 \qquad \forall k \in K, t \in T \tag{6}$$

$$\sum_{j\in N} x_{j0kt} \le 1 \qquad \forall k \in K, t \in T \tag{7}$$

$$s_{jkt} + S_j + \tau_{j0} \le D \qquad \forall j \in C, k \in K, t \in T \tag{8}$$

$$s_{jkt} - s_{ikt} \ge (\tau_{ij} + S_i)x_{ijkt} - D(1 - x_{ijkt}) \qquad \forall i, j \in C,\ i \ne j, k \in K, t \in T \tag{9}$$

$$etw_j \le s_{jkt} \le ltw_j \qquad \forall j \in C, k \in K, t \in T \tag{10}$$

$$x_{ijkt} \in \{0, 1\} \qquad \forall i, j \in N, k \in K, t \in T \tag{11}$$

$$z_{ikt} \in \{0, 1\} \qquad \forall i \in C, k \in K, t \in T \tag{12}$$

$$y_{it} \in \{0, 1\} \qquad \forall i \in C, t \in T \tag{13}$$

$$s_{ikt} \in \mathbb{R}_0^+ \qquad \forall i \in N, k \in K, t \in T \tag{14}$$

The objective function (1) minimizes transportation costs, inventory/waiting costs of customers served via the own fleet and backup option, and costs for customers served via the backup option. Constraints (2) ensure that every customer is delivered within the requested delivery interval via the own fleet or backup option. Constraints (3) prohibit the vehicle capacities to be exceeded. Constraints (4) are flow-conserving constraints. Constraints (5) link the assignment variables z_{ikt} with the flow variables x_{ijkt}. Constraints (6) and (7) ensure that only one tour is performed per vehicle and day. Constraints (8)–(10) guarantee that the vehicle arrives back at the depot in time, customer TWs are respected and subtours are eliminated. Constraints (11)–(14) define the respective domain of the variables.

Model SDDP can be used to calculate lower bounds for the stochastic-dynamic case, i.e., the costs that arise under the assumption of perfect information. However, model SDDP is difficult to solve for real-sized instances. For example, the related model by Archetti et al. (2015) can only consistently handle instances with fewer than 50 customers. We leave this open for future research and focus below on developing a solution approach that can be used to solve and analyze the stochastic-dynamic problem case.

5. Stochastic and dynamic solution approach

We use the deterministic model formulated as the basis for our stochastic-dynamic modeling approach. In this dynamic setting, only a subset of customer orders is known at planning instant, i.e., only customers revealing their demand on the current and previous days. The customer demands within the following days are uncertain and stochastic. New orders arrive at the end of each day. A periodic re-planning is carried out each day to account for the newly arrived orders and updated demand forecasts. The proposed solution approach is based on the suggested heuristics for a similar setting used by Albareda-Sambola et al. (2014). In this approach, we solve an auxiliary prize-collecting VRP with TWs (PCVRPTW, see Section 5.1) for the current day, that decides which customers to deliver on that day and the corresponding routing. We define a benefit measure for each known customer indicating the value of serving the customer on the current day rather than postponing the customer to future days. We henceforth call this benefit/reward measure "*prize*" following the nomenclature in vehicle routing literature (see e.g., Feillet et al., 2005). The *prize* combines positive and negative effects of servicing an order on the current day, including the following aspects: urgency of the order and the probabilities of emerging nearby customers on future days. As intuition, the prize should be high if the order is urgent or we expect a high demand on future days. Contrary, the prize should be low if we expect nearby customers to emerge in the coming days. We elaborate on the calculation of the prize in Section 5.2. The PCVRPTW is solved heuristically with a hybrid adaptive large neighborhood search with granular insertion operators (HALNS-G, see Section 5.3) that has shown promising results when solving several VRPs (Voigt et al., 2022, 2023).

5.1. Auxiliary prize-collecting vehicle routing problem with time windows

In this section, we model the PCVRPTW. The goal of the PCVRPTW is to maximize the sum of collected prizes minus the transportation costs for traveling to the customers to collect the prize during the current day, $t^{current}$. The set of customers known on this day is denoted by $C_{t^{current}}$. We formulate the problem with an equivalent minimization objective (15) which minimizes transportation costs and the sum of uncollected prizes, i.e., the sum of prizes for customers not served on this day. Ultimately, the decision variable v_i indicates whether we serve the customer immediately ($v_i = 1$, i.e., the prize is collected) or whether we postpone her/him to future days ($v_i = 0$).

Model PCVRPTW

$$Minimize \quad \sum_{k\in K}\sum_{i,j\in N} c_{ij}^{\text{trans}} \cdot x_{ijk} + \sum_{i\in C_{t^{current}}} c_i^{\text{prize}} \cdot (1 - v_i) \tag{15}$$

s.t.

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = v_i \qquad \forall i \in C_{t^{current}} \quad (16)$$

$$\sum_{i \in C_{t^{current}}} d_i \sum_{j \in N} x_{ijk} \leq Q \qquad \forall k \in K \quad (17)$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} \qquad \forall i \in N, k \in K \quad (18)$$

$$\sum_{j \in N} x_{0jk} \leq 1 \qquad \forall k \in K \quad (19)$$

$$\sum_{j \in N} x_{j0k} \leq 1 \qquad \forall k \in K \quad (20)$$

$$s_{jk} + S_j + \tau_{j0} \leq D \qquad \forall j \in C_{t^{current}}, k \in K \quad (21)$$

$$s_{jk} - s_{ik} \geq (\tau_{ij} + S_i)x_{ijk} - D(1 - x_{ijk}) \qquad \forall i, j \in C_{t^{current}},\ i \neq j, k \in K \quad (22)$$

$$etw_j \leq s_{jk} \leq ltw_j \qquad \forall j \in C_{t^{current}}, k \in K \quad (23)$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in N, k \in K \quad (24)$$

$$v_i \in \{0, 1\} \qquad \forall i \in C_{t^{current}} \quad (25)$$

$$s_{ik} \in \mathbb{R}_0^+ \qquad \forall i \in N, k \in K \quad (26)$$

Constraints (16) ensure that the prizes are collected only for customers delivered via the own fleet. Constraints (17) guarantee that the vehicle capacities are respected. Constraints (18) conserve flow. Constraints (19) and (20) ensure that only one tour is performed per vehicle. Constraints (21)–(23) guarantee that the vehicle arrives at the depot in time, customer TWs are respected and subtours are eliminated. Constraints (24)–(26) define the respective domain of the variables. As only small instances of this formulation can be solved with exact procedures, we propose a metaheuristic solution approach in Section 5.3.

5.2. Determining prizes

A crucial part of the overall solution approach is to appropriately determine the prize within the PCVRPTW, in order to make the right decision about which customers to serve immediately and which to postpone to future days. The calculation of the prize heavily depends on data availability; more information allows for more nuanced decision-making regarding which customers to serve on the current day and which to postpone. It is also conceivable that prizes could be determined using more sophisticated approaches, such as machine learning. Here, however, we present a straightforward and intuitive calculation of prizes, leaving potential improvements to future research. Nevertheless, please refer to Appendix A where we demonstrate a method for extending prize calculations to advanced scenarios.

In cases of relatively stable demand, a sufficient number of vehicles, and negligible inventory costs, we use only stochastic information on customer densities within the delivery area. Such information can be derived from, for example, publicly available population densities or the company's demand data. Let c_i^{prize} denote the prize of customer i that utilizes this stochastic information in Eq. (27).

$$c_i^{\text{prize}} = \begin{cases} c_i^{\text{back}} & \text{if } t_i^{\text{remaining}} = 0 \\ \omega \cdot \overline{c^{\text{trans}}} \cdot \frac{n_{i,t^{\text{current}}}}{\sum_{t=t^{\text{current}}}^{l_i} E[N_{it}]} & \text{if } t_i^{\text{remaining}} > 0 \end{cases} \qquad \forall i \in C_{t^{current}} \quad (27)$$

This measure primarily depends on the urgency, i.e., the remaining days $t_i^{\text{remaining}}$ to serve the customer. The remaining days are calculated as the difference between the latest delivery day l_i and the current day t^{current}, i.e., $t_i^{\text{remaining}} = l_i - t^{\text{current}}$. If the order is urgent (i.e., $t_i^{\text{remaining}} = 0$), the prize corresponds to the backup costs of the respective customer, c_i^{back}.

For postponable orders ($t_i^{\text{remaining}} > 0$), we define the prize as follows. Recall, that customers with high prizes (rewards) are more likely to be served on the current day. Intuitively, the prize should be higher when few days are remaining. Conversely, the prize should be low when there are only few customers nearby but more customers are expected to appear nearby in future days. More formally (see the second case of Eq. (27)), the prize depends on a term weighted by ω, an empirically determined factor that adjusts the prizes and guides decision-making. This term is calculated using the average transportation costs when all customers are served as early as possible $\overline{c^{\text{trans}}}$ and the number of nearby customers on the current day $n_{i,t^{\text{current}}}$ divided by the sum of expected nearby customers for customer i over the remaining delivery days. There may be several ways to estimate the expected transportation cost. In practice, one could use the historical average transportation costs (i.e., the total transportation costs over several periods divided by the number of customers served) or the costs incurred when serving all currently known customers in a single period (i.e., solving a classical VRP under an earliest policy). For our experiments, we use the average transportation cost for the respective instance class under the earliest policy.

The number of nearby customers $n_{i,t^{\text{current}}}$ counts how many customers are within an area A_i around customer i on the current day. A_i is defined as the area encompassing customers reachable from customer i within 10% of the average transportation cost, i.e., $0.1\overline{c^{\text{trans}}}$. For future days N_{it} is a random variable, that depends on the customer density around customer i within area A_i.

5.3. Hybrid adaptive large neighborhood search with granular insertion operators for the PCVRPTW

This section outlines the hybrid adaptive large neighborhood search with granular insertion operators (HALNS-G) and explains the problem-specific operators, in particular the granular insertion operators, which play a crucial role in improving the original HALNS as presented in Voigt et al. (2023). The concept of granular insertion operators is inspired by the granular tabu search (Toth & Vigo, 2003). This approach restricts the neighborhood of local search operators based on geographic proximity. In other words, it only considers moves that affect nearby edges. In the granular insertion operators suggested, the insertion positions are similarly confined to customers located in close proximity to the customer being inserted.

5.3.1. Framework

As the HALNS-G (see Algorithm 1) closely resembles the original HALNS, we provide a concise description only. For a more comprehensive understanding of this framework, we refer interested readers to the works of Voigt et al. (2022, 2023). The main distinction lies in the usage of granular insertion operators within the ALNS, in the following referred to as ALNS-G. The HALNS-G combines ideas from genetic algorithms, in particular the use of a population of individuals (solutions) and a crossover mechanism, with the well-known adaptive large neighborhood search (ALNS) introduced by Ropke and Pisinger (2006). Genectic algorithms construct solutions by iteratively generating new solutions from parent solutions with a crossover mechanism and potentially mutation operators and/or local search operators. In contrast, the ALNS is a neighborhood-based metaheuristics that generates solutions by iteratively removing customers from one solution and reinserting them by means of a removal and insertion operator, respectively. The operators are chosen from a set of available operators in an adaptive manner, i.e., the likelihood of selecting an operator depends on its performance during the search.

The HALNS-G starts by generating an initial population P of individuals by executing the ALNS-G n^{P} times (lines 1–3). This means each run of the ALNS-G generates a solution which is then added to the population of individuals. Subsequently, the ALNS-G is employed to create new individuals by comparing a solution from the population, denoted as s, with the global best solution, denoted as $\hat{s}$ (line 8–15). This crossover mechanism is based on the assumption that the optimal solution shares similarities with both s and, to a greater extent,

1 **while** $|P| < n^{\mathrm{P}}$ **do** // Initial population
2 $\quad s \leftarrow$ ALNS-G()
3 $\quad P \leftarrow P \cup \{s\}$
4 **end**
5 **while** *time limit not reached* **do** // Generations
6 $\quad \hat{s} \leftarrow$ DetermineBestSolution(P)
7 $\quad i \leftarrow 0$
8 $\quad$ **while** $i < n^{\mathrm{P}}$ **do** // Crossover and ALNS-G phase
9 $\quad\quad s \leftarrow P[i]$
10 $\quad\quad$ **if** *gens without improvement* mod $gen^{\text{new-inds}} \neq 0$ **then**
11 $\quad\quad\quad s \leftarrow$ ALNS-G($s, \hat{s}$)
12 $\quad\quad$ **else**
13 $\quad\quad\quad s \leftarrow$ ALNS-G()
14 $\quad\quad$ **end**
15 $\quad\quad P \leftarrow P \cup \{s\}$
16 $\quad\quad i \leftarrow i + 1$
17 $\quad$ **end**
18 $\quad P \leftarrow$ DiversityManagement(P) // Select survivors and manage diversity
19 **end**

Algorithm 1: Hybrid adaptive large neighborhood search with granular insertion operators

Input : Starting solution s, global best solution $\hat{s}$, parameters (χ_0, n_0, it^{stop}, α, β, p^{binom}, σ_1, σ_2, σ_3, n^{granular})
Output: Best solution s^*
1 $s^* \leftarrow s$
2 **while** *Iterations without improvement* $< it^{\mathrm{stop}}$ **do**
3 $\quad$ ChooseOperators()
4 $\quad C^{\mathrm{RC}} \leftarrow$ GetRemovalCandidates($s, \hat{s}, p^{\mathrm{binom}}$) // Determine removal candidates
5 $\quad (s^{\mathrm{new}}, C^{\mathrm{R}}) \leftarrow$ Remove($s, C^{\mathrm{RC}}, p^{\mathrm{binom}}$) // Removal operators (Section 5.3.3)
6 $\quad C^{\mathrm{R}} \leftarrow$ Sort(C^{R}) // Sorting operators (Section 5.3.4)
7 $\quad s^{\mathrm{new}} \leftarrow$ Insert($s^{\mathrm{new}}, C^{\mathrm{R}}, n^{\mathrm{granular}}$) // Granular insertion operators (Section 5.3.5)
8 $\quad$ **if** $f(s^{\mathrm{new}}) < f(s^*)$ **then**
9 $\quad\quad s^*, s \leftarrow s^{\mathrm{new}}$
10 $\quad\quad$ UpdateGranularity(n^{granular})
11 $\quad\quad$ **if** $f(s^*) < f(\hat{s})$ **then**
12 $\quad\quad\quad \hat{s} \leftarrow s^*$
13 $\quad$ **else if** $accept(f(s^{\mathrm{new}}), f(s^*), \alpha)$ **then**
14 $\quad\quad s \leftarrow s^{\mathrm{new}}$ // Simulated annealing
15 $\quad$ **end**
16 $\quad \alpha \leftarrow \alpha \cdot \beta$
17 $\quad$ UpdateALNSParameters(σ_1, σ_2, σ_3)
18 **end**

Algorithm 2: Adaptive large neighborhood search with granular insertion operators

with $\hat{s}$. Consequently, if customers are placed in the same position in both solutions, it is presumed that this placement is likely to be present in the optimal solution as well and should therefore not be removed. We define a customer to be placed in the same position in both solutions in case its preceding and succeeding node are the same in both solutions. As such, the crossover mechanism aims at reducing the probability of removing such probably already well-placed customers. To diversify the population, the HALNS-G generates new individuals every $gen^{\text{new-inds}}$ generations without any improvement (line 13). In this case, the ALNS-G is applied in the same manner as in the first generation, i.e., without a starting solution from the population and without the global best solution. The individuals that are carried over to the next generation are selected based on their objective value and diversity (line 18). The HALNS-G terminates when a specified time limit is reached (line 5).

5.3.2. Adaptive large neighborhood search with granular insertion operators

Algorithm 2 describes the ALNS-G adapted for the PCVRPTW used within the HALNS-G framework, as depicted in the previous section.

In all but the first generation and after $gen^{\text{new-inds}}$ generations without improvement the ALNS-G takes a starting solution s and the global best solution $\hat{s}$ as input. In contrast, in the first generation, s is initialized by applying one of the insertion operators randomly chosen. Furthermore, several parameters (χ_0, n_0, it^{stop}, α, β, p^{binom}, σ_1, σ_2, σ_3, n^{granular}) need to be set appropriately. The values chosen and their meaning can be found in Appendix B.

The algorithm starts by initializing the local best solution s^* with the starting solution s (line 1). The ALNS-G iteratively seeks improvements until a stopping condition is met (line 2). During each iteration, operators (removal, sorting and insertion) are selected depending on their historical performance (line 3). Customers are added to the set of removal candidates C^{RC} under two conditions: (1) when their position in the current solution s differs from the global best solution $\hat{s}$ or (2) when they are randomly selected (line 4). To be more precise, for condition (1), a customer is added to C^{RC} if its direct predecessor or successor in s differs from the predecessor/successor in $\hat{s}$. For condition (2), a customer whose placement in s matches that in $\hat{s}$ (i.e., it has the same predecessor and successor) is added to C^{RC} with a probability p^{binom}. Only customers in the set C^{RC} can be removed from the solution using removal operators (line 5) and are added to the set of removed customers C^{R}. The number of customers to be removed is sampled in every iteration from a binomial distribution. The parameters for this distribution are set as $|C^{\mathrm{served}}|$, indicating the number of served customers in solution s (i.e., the customers that are served by a vehicle) and probability p^{binom}. The set of removed customers, which also represents the customers to be inserted, is then sorted by a sorting operator (line 6) to determine the insertion order. The selected insertion operator is subsequently applied to insert customers from C^{R} in the determined order. In contrast to the original HALNS, not all customers in C^{R} have to be reinserted. Customers remaining in the set C^{R} are accounted for with their uncollected prizes in the objective function. The ALNS-G uses simulated annealing as acceptance criterion (lines 8–14), where worse solutions are accepted with a probability depending on the difference of the objective values $f(s^{\mathrm{new}}) - f(s^*)$ and a decreasing temperature α. The initial temperature is determined instance-specific by using $\alpha = -\frac{\overline{\Delta E}}{\ln(\chi_0)}$, following the formula from Johnson et al. (1989). Here, $\overline{\Delta E}$ represents an estimate of the increase in cost for strictly positive transitions, and χ_0 is a parameter indicating the likelihood of accepting a worsening solution. To generate these transitions, we perform n_0 iterations of the ALNS-G. Whenever a new best solution s^* is found, the granularity sets are updated (line 10), as described in the following paragraph on granular insertion operators. Finally, the temperature α is reduced by multiplying it with a cooling rate β (line 16), and the algorithm's ALNS parameters are updated (line 18, see e.g., Ropke & Pisinger, 2006).

5.3.3. Removal operators

We use four removal operators, each named in accordance with the nomenclature suggested in Voigt (2025). All removal operators work with the set C^{RC}. Therefore, only customers included in C^{RC} may be removed from the solution.

- **Random customers:** Randomly removes customers from the solution.
- **Worst cost customers:** Removes customers in descending order of $\Delta_i - c_i^{\min}$. Δ_i denotes the change in transportation costs if customer i is removed from the solution, and $c_i^{\min}$ denotes the minimum insertion cost of customer i encountered during the search.

- **A posteriori score related customers - to seed customer:** The first customer (i.e., the seed customer) is removed using the aforementioned *random customers* operator. Subsequently, additional customers are selected for removal from the set C^{RC} based on a relatedness measure. This relatedness $Rel(l,m)$ between two customers l and m is measured by a score that considers both geographical proximity and the start times of service of the vehicle at each customer's location in the current solution. The relatedness score is calculated as follows: $Rel(l,m) = \frac{c_{lm}}{\max_{i,j \in N} c_{ij}} + \frac{|s_{l,k(l)} - s_{m,k(m)}|}{ltw_0}$
- **All customers - from randomly selected sequence within concatenated routes:** This operator concatenates all routes into one giant tour by randomly adding one tour after the other, and then randomly selects a customer to remove. It continues removing the successors of the selected customers if they are included in C^{RC} until the desired number of customers has been removed.

5.3.4. Sorting operators

The set of removed customers C^{R} is sorted with one of the following two sorting operators selected adaptively. These sorting criteria differ from the original proposed sorting operator. In the context of the PCVRPTW, the goal is not to serve every customer (as in e.g., CVRP or VRPTW), and as such priority is given to customers with high prizes.

- **Customers with highest ratio of prize and demand first:** This operator sorts customers in descending order of the ratio of their prize and demand, denoted as c_i^{prize}/d_i. Customers with a high prize/demand ratio are inserted earlier, increasing their chances of being delivered.
- **Customers with highest profit first:** This operator sorts customers in descending order of their profit. The profit of a customer is defined as the difference of their prize and the insertion costs, represented as $c_i^{\text{prize}} - \Delta_i$, where Δ_i denotes the change in transportation costs when customer i is present in the solution.

5.3.5. Granular insertion operators

After sorting C^{R} as described previously, the customers are inserted using granular insertion operators. The goal of the granular insertion operators is to determine the most suitable positions to insert customers into the existing solution. This operation is based on the concept of *insertion operator - best cost position* (Voigt, 2025). Instead of considering all possible insertion positions, the granular approach narrows the search scope, focusing on insertion positions that are more likely to yield improved solutions. The classical *insertion operator - best cost position* typically iterates across all vehicles and nodes to find the position where the customer can be inserted with lowest total cost. In the granular implementation, the operator does not iterate across all nodes but restricts the search to nodes that are likely to appear as predecessors of the customer to be inserted. This significantly reduces run time without being detrimental to solution quality. Crucial for this presumption to be true is an appropriate restriction of nodes to examine. Let N_j^{granular} denote the granular node set including only nodes that we deem likely to appear as predecessor of customer j i.e, nearby nodes or nodes that have been a predecessor in previous good solutions. In more detail, we determine $N_j^{\text{granular}} = N_j^{\text{sorted}}[1,\ldots,n^{\text{granular}}]$ where N_j^{sorted} denotes the node set sorted in increasing order of c_{ij}^{trans} and n^{granular} the number of nodes to be examined. In other words N_j^{granular} initially includes the n^{granular} nearest neighbors of customer j. The set N_j^{granular} is dynamically updated when a new best solution s^* is found such that if node i is followed by customer j in s^*, all nodes having lower or equal c_{ij}^{trans} to node i are now included in N_j^{granular}. More formally, we set $n^{\text{granular}} = \max(n^{\text{granular}}, \text{index of node } i \text{ in } N_j^{\text{sorted}})$.

To account for the possibility that not all customers must be served in the solution, the customer will only be inserted if the penalized insertion costs are lower than a given acceptance threshold. The insertion costs are the additional routing costs incurred when adding the customer to the route. These costs may be penalized by a factor when the insertion would lead to violating time windows or capacity constraints. The threshold is sampled from a uniform distribution ranging from $[min, max]$ times the average costs per customer in the current solution s. There are two variants of the granular best position insertion operator based on the acceptance threshold.

- **High acceptance threshold:** This operator is designed to potentially insert more customers into the solution. The threshold is sampled from a uniform distribution ranging from min^{high} to max^{high}. Therefore, it accepts to insert customers with insertion costs that are (up to max^{high} times) higher than the average costs per customer, meaning that customers with relatively high costs are still inserted. This approach aims to diversify the search.
- **Low acceptance threshold:** In contrast, this operator uses a lower acceptance threshold. The threshold is sampled from a uniform distribution ranging from min^{low} to max^{low}, with $min^{\text{low}} < max^{\text{low}} < min^{\text{high}} < max^{\text{high}}$. Consequently, only customers with low insertion costs are accepted. This approach is more selective and tends to intensify the search.

6. Numerical experiments

In this section, we present results of several numerical experiments. In Section 6.1 we establish that the HALNS-G works well on PCVRPTW instances and analyze how the granular insertion operators impact the overall performance of the HALNS. In Section 6.2 we introduce the simulation framework and use it to conduct several experiments aimed at generating valuable managerial insights.

The HALNS-G is implemented in C++ and run on an AMD Ryzen 7 2700X with 32 GB RAM. The simulation component is implemented using Python 3.7. Parameters of HALNS-G can be found in Appendix B. These values are set according to previous research and preliminary experiments.

6.1. Performance evaluation of HALNS-G

6.1.1. Performance on PCVRPTW instances with 1000 customers

We evaluate the performance of the HALNS-G on PCVRPTW instances with 1000 customers based on VRPTW instances by Homberger and Gehring (1999) against pyVRP (Wouda, Lan et al., 2024) in Table 3. The instances are developed by Wouda, Aerts-Veenstra et al. (2024) and can be found at PyVRP (2023). Both algorithms are run on the same machine with the same time limits.

pyVRP is a powerful improved version of the open-source hybrid genetic search introduced by Vidal (2022) which has consistently shown outstanding performance in implementation challenges for solving VRPs (http://dimacs.rutgers.edu/programs/challenge/vrp/) and dynamic VRPTWs (https://euro-neurips-vrp-2022.challenges.ortec.com/). Furthermore it has a direct implementation for the PCVRPTW developed for solving a waste-collection problem (Wouda, Aerts-Veenstra et al., 2024) and as such no adjustments were necessary for our benchmarks. Table 3 shows the instance names, best-known solutions (BKS, available on the pyVRP GitHub repository), the respective objective values and the percentage gap to the BKS ($Gap = \frac{Objective - BKS}{BKS}$) for both pyVRP and HALNS-G, respectively. Instances labeled with C represent cases with clustered customers, R with randomly distributed customers, and RC instances feature a mix of randomly distributed and clustered customers. Upon examining the average gap (last row), it becomes evident that both HALNS-G and pyVRP yield similar gaps compared to the BKS.

Table 3
Results for PCVRPTW instances with 1000 customers.

Instance	BKS	pyVRP		HALNS-G	
		Objective	Gap [%]	Objective	Gap [%]
C1_10_1	24 539.1	24 559.5	0.08	24 539.1	0.00
C1_10_2	24 901.5	24 913.7	0.05	**24 848.7**	**−0.21**
C1_10_3	24 502.8	24 419.3	−0.34	**24 410.7**	**−0.38**
C1_10_4	24 456.2	24 464.5	0.03	**24 410.4**	**−0.19**
C1_10_5	24 860	24 871.8	0.05	**24 857.3**	**−0.01**
C1_10_6	24 676.3	24 718.1	0.17	**24 652.4**	**−0.1**
C1_10_7	24 463.1	24 469.6	0.03	24 463.4	0.00
C1_10_8	24 427.3	24 423.2	−0.02	**24 423.2**	**−0.02**
C1_10_9	24 589.1	24 609.9	0.08	24 603.2	0.06
C1_10_10	24 860.5	24 830.4	−0.12	**24 808.6**	**−0.21**
C2_10_1	16 581.6	16 636	0.33	16 613.9	0.19
C2_10_2	16 212.8	16 292.1	0.49	16 225.3	0.08
C2_10_3	15 718.4	15 846.5	0.81	15 837.4	0.76
C2_10_4	15 096.9	15 524.9	2.84	15 449.4	2.33
C2_10_5	16 257.1	16 291	0.21	16 300.5	0.27
C2_10_6	16 004.3	16 167.1	1.02	16 085.8	0.51
C2_10_7	16 017.7	16 115.3	0.61	16 031.7	0.09
C2_10_8	15 706.7	15 817.3	0.7	15 812.6	0.67
C2_10_9	15 819.2	15 997.4	1.13	15 857.7	0.24
C2_10_10	15 362.5	15 535.7	1.13	15 519	1.02
R1_10_1	26 271.1	26 274	0.01	26 271.1	0.00
R1_10_2	26 312.5	26 314.4	0.01	26 321.7	0.03
R1_10_3	25 615.7	25 617.8	0.01	25 657.6	0.16
R1_10_4	25 212.2	25 219.5	0.03	25 251.1	0.15
R1_10_5	25 788.2	25 789.9	0.01	25 790.1	0.01
R1_10_6	25 397	25 401.7	0.02	25 406.5	0.04
R1_10_7	25 137.8	25 142.3	0.02	25 152.2	0.06
R1_10_8	24 896	24 904.9	0.04	24 931.1	0.14
R1_10_9	25 433.4	25 433.4	0.00	25 443.5	0.04
R1_10_10	25 790.3	25 790.3	0.00	25 816.4	0.1
R2_10_1	24 115.9	24 200.4	0.35	**24 072.5**	**−0.18**
R2_10_2	20 975.8	21 205.1	1.09	21 081.1	0.5
R2_10_3	18 269.5	18 605.4	1.84	18 530.8	1.43
R2_10_4	15 689.5	15 910.1	1.41	15 880.4	1.22
R2_10_5	23 461.4	23 599.9	0.59	**23 452.6**	**−0.04**
R2_10_6	20 423.1	20 485.8	0.31	20 468.7	0.22
R2_10_7	17 790.1	18 167.2	2.12	17 852.5	0.35
R2_10_8	15 542.4	15 782	1.54	15 645.2	0.66
R2_10_9	22 990.7	23 019.6	0.13	22 997	0.03
R2_10_10	22 133	22 322.2	0.85	22 282.3	0.67
RC1_10_1	24 816.1	24 826.1	0.04	24 817.3	0.00
RC1_10_2	25 043.1	25 047.4	0.02	25 066.4	0.09
RC1_10_3	24 461.4	24 477.6	0.07	24 488.4	0.11
RC1_10_4	24 495.6	24 523.1	0.11	24 528.8	0.14
RC1_10_5	25 102.6	25 113.3	0.04	25 126.6	0.1
RC1_10_6	24 623.5	24 625.7	0.01	24 650.3	0.11
RC1_10_7	24 874.2	24 894.3	0.08	24 886.4	0.05
RC1_10_8	24 549.7	24 565.6	0.06	24 559.3	0.04
RC1_10_9	24 435.3	24 488	0.22	24 482.9	0.19
RC1_10_10	24 615.2	24 634.5	0.08	24 653.7	0.16
RC2_10_1	19 726.8	19 863.7	0.69	19 746	0.1
RC2_10_2	17 359.1	17 635.1	1.59	17 623.8	1.52
RC2_10_3	15 419.6	15 562.4	0.93	15 683.2	1.71
RC2_10_4	13 917.6	14 296.8	2.72	14 092.7	1.26
RC2_10_5	18 198.7	18 412.9	1.18	18 427	1.25
RC2_10_6	18 257	18 448	1.05	18 315	0.32
RC2_10_7	17 718.4	17 852.8	0.76	17 827.2	0.61
RC2_10_8	17 046.5	17 113.4	0.39	17 164.5	0.69
RC2_10_9	16 576	16 739.3	0.99	16 826.1	1.51
RC2_10_10	16 625.9	16 779.8	0.93	16 779.2	0.92
Average	21 336.0	21 426.5	0.53	21 396.7	0.36

Both algorithms have been run with a time limit of 60 min.

6.1.2. Performance on PCVRPTW instances with increasing number of customers

To evaluate the scalability of HALNS-G as the number of customers increases, we have created a set of instances featuring 200, 400, 600, and 800 customers by randomly sampling the desired number of customers from the 60 instances with 1000 customers. This leads to an additional 240 instances. The performance of HALNS-G compared to pyVRP across various instance sizes is summarized in Table 4 ($Gap = \frac{Objective^{\text{HALNS-G}} - Objective^{\text{pyVRP}}}{Objective^{\text{pyVRP}}}$). The HALNS-G tends to outperform pyVRP on medium-sized instances. This observation is particularly relevant as we execute the simulation (see Section 6.2) with instances featuring a moderate number of customers. Consequently, HALNS-G appears to be a suitable choice for solving such instances.

Table 4
Average gap against pyVRP [%] for PCVRPTW instances with $200, 400 \ldots, 1000$ customers.

Instance	Number of customers				
	200	400	600	800	1000
C1	0.00	0.00	0.00	−0.06	−0.11
C2	−0.17	−0.53	−1.15	−0.15	−0.31
R1	0.00	0.00	0.00	0.03	0.06
R2	−0.07	−0.44	−0.54	−0.56	−0.53
RC1	0.00	0.00	−0.01	−0.01	0.03
RC2	−0.16	−0.74	−0.81	−0.34	−0.13
Average [%]	−0.07	−0.28	−0.42	−0.18	−0.16
Runtime [min]	6	12	30	45	60

Table 5
Average gap against pyVRP [%] for PCVRPTW instances with 200 customers and varying prizes.

	Multiple of original prize				
	x1	x2	x3	x4	x5
Average [%]	−0.07	−0.01	0.27	0.57	0.47

Both algorithms have been run with a time limit of 6 min.

6.1.3. Performance on PCVRPTW instances with 200 customers and increasing prizes

The prizes applied in the PCVRPTW can vary greatly depending on the urgency of a customer delivery. We therefore evaluate the robustness of the HALNS-G across a range of prizes by varying the prizes in instances with 200 customers by multiplying the original prize with a factor ranging from 1 to 5. Table 5 shows once more the average gap of the HALNS-G against pyVRP. The HALNS-G exhibits superior performance on instances with lower prizes (factors 1 to 2) when compared to pyVRP. In contrast, for instances with higher prize values (factors of 3 to 5), pyVRP attains better solutions. In these scenarios, routing decisions become more challenging as a larger number of customers should be served. Given that pyVRP is built on the hybrid genetic search and incorporates a highly effective local search capable of identifying near-optimal routing solutions, the observed results align with expectations.

6.1.4. Contribution of granular insertion operators

This section examines the contribution of granular insertion operators on the overall performance. We implement the original HALNS, which does not restrict the set of insertion positions, i.e., $N_j^{\text{granular}} = N, \forall j \in C$. We then compare the performance of this HALNS with regular insertion operators (HALNS) against HALNS-G on PCVRPTW instances with $200, \ldots, 1000$ customers. Fig. 2 shows the average gaps against pyVRP for the various instance sizes, achieved without (HALNS) and with granular insertion operators (HALNS-G).

HALNS-G clearly outperforms HALNS, and the effect becomes more pronounced as the number of customers increases. In fact, the granular insertion operators are responsible for achieving superior performance for instances with 600 and more customers against pyVRP. This suggests that granular insertion operators may be a powerful and straightforward device for improving the performance of the ALNS methodology in general.

6.2. Experiments and managerial insights

The experiments aim to quantify the benefits associated with introducing a some-day option compared to an earliest policy (EP) which may be followed in practice. In the EP, each customer is served as

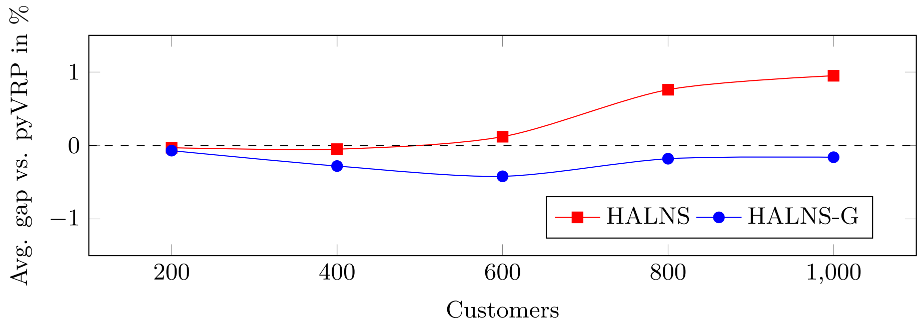


Fig. 2. Effect of granular insertion operators (HALNS vs. HALNS-G).

quickly as possible, i.e., within the day following the order. Section 6.2.1 introduces the simulation framework. Section 6.2.2 investigates the relationship between the length of the some-day option and the achievable savings in costs and vehicles. Section 6.2.3 examines the savings associated with increasing the share of customers choosing the some-day option. Section 6.2.4 investigates whether a retailer can offer customers the option to select TWs in exchange for accepting an extended delivery interval.

6.2.1. Simulation framework and performance measures

The simulation is based on six VRPTW instances by Homberger and Gehring (1999) with 1000 customers (C1_10_1, C2_10_1, R1_10_1, R2_10_1, RC1_10_1, and RC2_10_1, see Table 3). For each day, the simulator randomly samples a set of 100 customers from the respective instance. Each customer is characterized by its coordinates, demand and TW, as specified in the original instance. This set is then revealed as the pending customer set for the current planning day. We assume no inventory or waiting costs, sufficiently high backup costs and an unlimited number of vehicles in order to achieve meaningful and, above all, comparable results between the different experiments. The backup costs for customers ($c_j^{\text{back}} > c_{0j} + c_{j0}$) are set so high that they are served with the given fleet in any case, even if this requires a separate vehicle for an individual customer. Consequently, each customer can be served within its delivery day interval. We are interested in the steady state (long-term) behavior of the system. To achieve this, we exclude the first five days and the last day, as these days are not representative of the system's long-term behavior. For each instance, we generate 30 days, thus providing 24 days for calculating three performance measures: (1) Average costs per customer served, calculated as the total transportation costs divided by the number of customers served, (2) average number of vehicles used per day, and (3) average delay per customer who have chosen the some-day option.

6.2.2. Impact of delivery interval length of some-day option

In this experiment we analyze the impact of the interval length of the some-day option. We assume that all customers select the some-day option, and we increase the delivery interval in each simulation by one day up to a length of five days. The goal of this experiment is to quantify the savings in costs and number of vehicles used when compared to the EP. Note that the EP is equivalent to the case where the delivery interval length is zero days, i.e., $l_i = e_i,\ \forall\, i \in C$.

We analyze two distinct scenarios. In one case, we assume that the logistics service provider must respect TWs (*with TWs*), meaning customers are only available to receive their deliveries within specific time windows during the day. The other case assumes no predetermined TWs for the delivery day (*without TWs*), allowing customers to receive their deliveries at any time throughout the selected day. For example, in the *with TWs* case, the shipping of high-value products or bulky goods might require deliveries at certain times to align with the availability of the customer. In contrast, general parcel deliveries, may not require specific TWs (*without TWs*). In both cases, however, it is possible to select the day of delivery from an interval of several days (some-day option).

Fig. 3 presents the results of the simulations. The upper plot shows the scenarios where customer TWs are respected, and the lower plot the scenarios without customer TWs. Both plots display the delivery interval length on the x-axis ($l_i - e_i,\ \forall\, i \in C$), the average number of vehicles and the average costs on the y-axis, and the average delay on the secondary y-axis. The average number of vehicles and the average costs are standardized against the EP. Following the EP, the scenarios analyzed show that incorporating TWs results in average costs per customer of 96.34 compared to 48.30 when TWs are not considered. Similarly, the average vehicle requirements are 15.5 with TWs and 6.1 without. Consequently, when adherence to TWs is required, the some-day option can be expected to yield a greater savings potential compared to scenarios without TW constraints.

In both cases (with and without customer TWs), the average costs and average number of vehicles used decrease significantly with an increasing length of delivery intervals. However, the savings become increasingly marginal with every additional day. The actual delay is far less than the given delivery interval length, for instance customers can expect to be served within 1.67 days (with TWs) and 2.18 days (without TWs) even if they have chosen the some-day option with 5 days, respectively. Interestingly, the reduction in costs goes along with the reduction in vehicles only if TWs are respected (upper plot). For scenarios without TWs (lower plot), the number of vehicles is only slightly affected by the delivery interval, but the transportation costs are. This means that the distance per vehicle is noticeably reduced, but not the number of vehicles used.

In summary, the results show that the some-day option offers a relatively higher savings potential in scenarios where specific TWs have to be respected on the delivery day compared to scenarios where no specific TWs have to be taken into account. Independent of the presence of TWs, a delivery interval length of three days seems sufficient to reduce costs up to 60% compared to the EP in exchange for a moderate increase in delay of less than 1.5 days. As an added bonus, the average number of vehicles is decreased, in particular when TWs are present. This is especially important in the light of driver shortages.

6.2.3. Impact of share of customers selecting the some-day option

If the retailer/service provider offers a range of delivery options with varying speeds, from which the customer can freely choose, it is obvious that not all customers will opt for the some-day option. To account for these scenarios, we vary the share of customers selecting the some-day option and analyze its impact on the same measures as shown before. In this experiment, we randomly sample a share of customers to select the some-day option (with a length of three days), while all other customers are assumed to select the fastest delivery option. We incrementally increase the share from 10% to 100% in steps of 10% in Fig. 4 for the case with TWs. Similar results are achieved for the case without TWs.

The figure shows nearly linear relations between the share of customers selecting the some-day option and the performance measures.

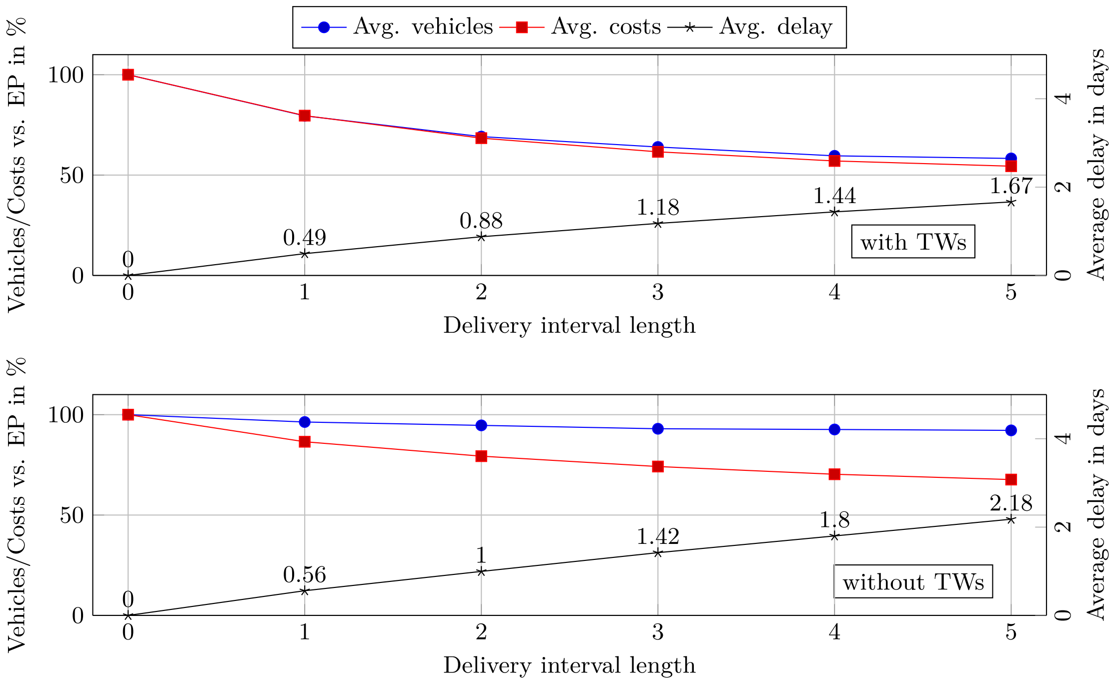


Fig. 3. Effect of delivery interval length.

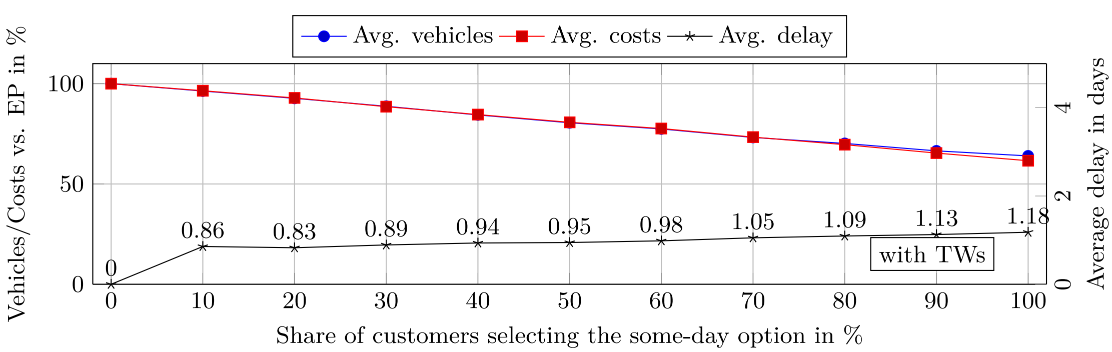


Fig. 4. Effect of share of customers selecting the some-day option for the case with customer time windows.

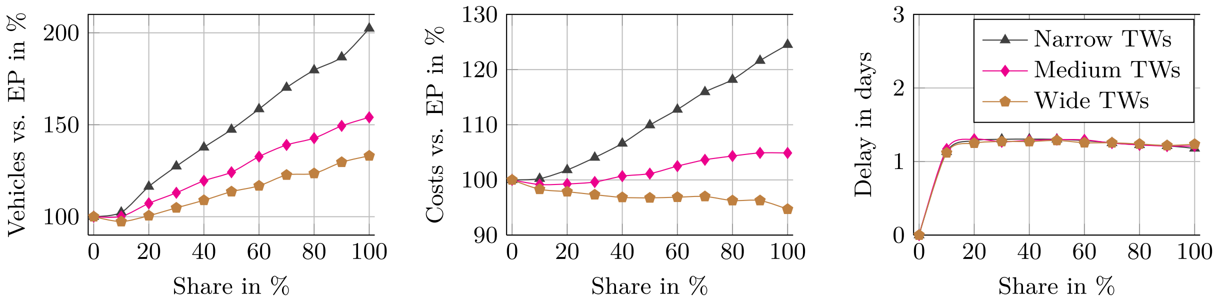
If the retailer motivates an additional 10% of customers to choose the some-day option, costs are expected to reduce by 3.9%, and the number of vehicles by 3.7%. These analyses may help estimate the benefits (e.g., discounts on shipping costs) the retailer may offer their customers in exchange for their willingness to wait. Furthermore, the experiment shows that cost savings are possible with a small number of customers opting for the some-day option. This can be achieved without additional costs for the retailer, for instance, by showcasing the emissions savings associated with selecting the some-day option (Dietl et al., 2024). Due to the linearity of the reduction in transportation costs, the retailer could provide a rough estimate of emission savings. However, to produce a precise estimate of achievable emissions savings, additional factors need to be considered, such as vehicle load, elevation profile, and vehicle type. Consequently, further research is required to accurately quantify the potential impact on emissions.

6.2.4. Impact of a some-day option with the incentive of selecting time windows

Retailers may consider increasing the attractiveness of the some-day delivery option by allowing the customers to choose a delivery TW. In that case customers get in exchange for their patience the option to select their preferred delivery time. This is particularly appealing for customers with a predictable schedule, such as an employed person available after work every day. Such an option can also benefit the service provider by reducing failed delivery attempts (see Voigt et al., 2023), as customers can choose a time when they are more likely to be at home. However, there is a downside to the introduction of TWs, as it significantly increases the number of vehicles required and transportation costs, as illustrated by Punakivi and Saranen (2001) and by our experiments in Section 6.2.2 (see above). This experiment aims to determine if the some-day option may compensate for these negative effects. For this purpose, we once again vary the share of customers selecting the some-day option (with 3 days) in Fig. 5.

TWs are exclusively applicable for customers who have chosen the some-day option. We benchmark against the EP without TWs. Additionally, we vary the width of the TWs. The original TW width given in the respective instance is considered as *narrow TWs*. In the original instance, the TW width averages 2% of the delivery horizon equivalent to a mere 15-min time frame assuming a 12-h delivery horizon. *Medium TWs* are four times wider (on average approx. 1 h), and *wide TWs* are eight times wider (on average approx. 2 h).

Fig. 5(a) shows that the number of vehicles heavily increases with the share of customers choosing the some-day option with TWs as



(a) Average vehicles (b) Average costs (c) Average delay

Fig. 5. Effect of share of customers selecting the some-day option when time windows with different widths are given as incentive compared to the earliest policy (EP) without any time windows.

incentives. However, the magnitude of the increase in vehicles is dependent on the width of the TW. Notably, only for wide TWs and a low share of customers choosing the some-day option, a slight decrease in vehicles is achievable (see the brown line with pentagon markers at a 10% share).

The development of costs (Fig. 5(b)) is heavily dependent on the TW width. With narrow TWs (dark gray line with triangle markers), costs steadily increase but at a moderate pace. Therefore, implementing narrow TWs as an incentive for an extended delivery interval may only be sensible in terms of costs when the number of failed deliveries due to customer absence is high and costly. For medium sized TWs (magenta line with diamond markers), costs decrease only when up to 30% of customers select the some-day option with these medium-sized TWs as incentive. When more customers demand TWs, the some-day option cannot fully compensate for the additional costs associated with TWs. The retailer/service provider can take advantage of these results by dynamically adjusting the offer. They can suspend the offer as soon as the number of some-day customers reaches the break-even point and resume it when it becomes profitable again. The situation changes when TWs are wide (brown line with pentagon markers). In this case, as more customers choose the same-day option, costs steadily decrease. However, the incentive (i.e., selecting TWs) for a some-day option becomes less attractive with wider TWs.

The average delay per customer (Fig. 5(c)) remains unaffected by the width of the TW and the share of customers selecting the some-day option with TW incentive. Thus, during checkout, the retailer can inform the customer about the expected average delay without needing to know the exact share of customers selecting the some-day option.

In summary, this experiment clearly shows that the retailer/service provider must carefully evaluate which width of TWs is attractive for customers, anticipate the share of customers choosing the some-day option with the given TW incentive, and weigh the reduction in transportation costs (plus failed delivery costs) against the probable increase in vehicles.

7. Conclusion

Summary. This paper is motivated by the idea to apply the concept of shipment consolidation over time from the field of slow logistics to e-commerce deliveries. The potential for increased consolidations is achieved by providing the customer with a slow delivery alternative, termed some-day option. Customers may opt for this alternative because they receive monetary incentives or additional perks from the retailer or parcel service provider. It may also be sufficient to simply inform customers of the environmental impact of immediate delivery and they may choose the some-day option due to their environmental awareness. This gives the retailer or the parcel service provider more flexibility when to deliver the customer and therefore more opportunities to bundle shipments from the customer's neighborhood over time, resulting in cost savings and lower emissions. After establishing the basic slow delivery concept, we identified the required features and cost components for a dynamic-stochastic optimization model considering the last mile delivery. In a first step, we model the entire decision problem as a multi-period VRP and assume a static and deterministic setting for definition purposes. For the dynamic and stochastic setting, a solution approach is then implemented based on solving auxiliary prize-collecting VRPs with time windows (PCVRPTW). The PCVRPTW in turn, is solved by a tailored hybrid adaptive large neighborhood search with granular insertion operators (HALNS-G). The HALNS-G outperforms state-of-the-art metaheuristics, when the trade-off between transportation costs and prizes is more pronounced. Our experiments show that significant cost savings are possible with a moderate increase in delay. Furthermore, the savings exhibit a linear relation to the share of customers choosing the some-day option. This means that the retailer benefits, even if only a minority of customers can be motivated to choose the some-day option. The simulation also shows, that time windows as incentive for selecting the some-day option can be reasonable, if the retailer expects high costs resulting from failed deliveries, and/or the time windows are sufficiently wide.

Limitations and future areas of research. In addition, we would like to draw attention to some interesting research directions that can be examined with regards to the problem setting and solution approach we have presented in this article. In terms of sustainability, we have thus far quantified only the economic benefits of the some-day option and anticipate potential reductions in emissions. Further research should focus on precisely quantifying these emission reductions, factoring in variables such as vehicle types, speed, elevation, and load (e.g., Rave & Fontaine, 2024). Additionally, the social impact of workload balancing deserves further investigation. This article primarily addresses routing within a B2C context. However, the some-day option could also enhance picking and packing processes by allowing multiple orders from the same customer to be consolidated, thus further streamlining logistics. Moreover, the slow logistics concept could be adapted for application in business-to-business (B2B) environments. Regarding our solution approach, future research might explore more sophisticated methods for prize calculation within the prize-collecting VRP, potentially leveraging machine learning or other advanced techniques to set appropriate prizes. Our experiments demonstrate that HALNS with granular insertion operators is effective. This suggests that studying the impact of granular insertion operators within ALNS across other problem types could yield valuable insights.

CRediT authorship contribution statement

Stefan Voigt: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Markus Frank:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Heinrich Kuhn:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Conceptualization.

Acknowledgments

We thank the editor and the anonymous reviewers whose constructive comments helped us to significantly improve the paper.

Appendix A. Prize calculations for scenarios with unstable demand, limited number of vehicles or inventory costs

For our numerical experiments (see Sections 5.2 and 6.2), we assumed a stable demand, a sufficient number of vehicles and no inventory costs. In this section, we demonstrate how the prize calculation could be easily adapted to relax these assumptions.

Intuitively, the prizes should be higher when inventory costs are significant and future capacity utilization is expected to be high. The prizes can be calculated by using Eq. (27) with additional terms for the case where the order can still be postponed ($t_i^{\text{remaining}} > 0$).

In the presence of inventory costs, incurred for each day the order is postponed, these costs are simply added to the prize. As a result, the order will only be postponed if the expected savings exceed the inventory costs.

$$c_i^{\text{prize}} = \begin{cases} c_i^{\text{back}} & \text{if } t_i^{\text{remaining}} = 0 \\ \omega \cdot \overline{c^{\text{trans}}} \cdot \frac{n_{i,t^{\text{current}}}}{\sum_{t=t^{\text{current}}}^{l_i} E[N_{it}]} + c_i^{\text{inv}} & \text{if } t_i^{\text{remaining}} > 0 \end{cases} \quad \forall i \in C_{t^{current}} \tag{A.1}$$

In order to account for unstable demand (or a varying number of vehicles each day), we need forecasts of total demands in the upcoming days (and/or the number of available vehicles) to estimate the expected capacity utilization in the remaining delivery days, $E[U_i]$. This measure depends on the number of the currently known customers, $|C_{t^{\text{current}}}|$ and the expected number of customers in the following remaining days, $E[|C_t|]$, $t = t^{\text{current}} + 1, \dots, t_i^{\text{remaining}}$, divided by the maximum possible number of customers served in the remaining days. The latter number is quantified by the available vehicles $|K_t|$ in the remaining planning horizon, $t = t^{\text{current}}, \dots, t_i^{\text{remaining}}$ and the expected number of customers served per vehicle $E[n^{\text{vehicle}}]$

$$E[U_i] = \frac{|C_{t^{\text{current}}}| + \sum_{t=t^{\text{current}}+1}^{l_i} E[|C_t|]}{E[n^{\text{vehicle}}] \cdot \sum_{t=t^{\text{current}}}^{l_i} |K_t|} \tag{A.2}$$

This factor is then incorporated into the prize calculation given in Eq. (A.1) as follows:

$$c_i^{\text{prize}} = \begin{cases} c_i^{\text{back}} & \text{if } t_i^{\text{remaining}} = 0 \\ \omega \cdot \overline{c^{\text{trans}}} \cdot E[U_i] \cdot \frac{n_{i,t^{\text{current}}}}{\sum_{t=t^{\text{current}}}^{l_i} E[N_{it}]} & \\ \quad + c_i^{\text{inv}} & \text{if } t_i^{\text{remaining}} > 0 \end{cases} \quad \forall i \in C_{t^{current}} \tag{A.3}$$

Appendix B. Parameters

See Table B.6.

Table B.6
Parameters for HALNS-G.

Parameter	Meaning	Chosen value
n^{P}	Size of the initial population	12
$gen^{\text{new-inds}}$	Number of generations without improvement	5
it^{stop}	Number of iterations without improvement (one ALNS run)	5000
p^{binom}	Probability for binomial distribution drawn for every ALNS run	$unif(0.12, 0.24)$
β	Cool rate in simulated annealing	0.9999
χ_0	Acceptance probability in simulated annealing	0.04
n_0	Number of iterations for determining the initial temperature	400
σ_1	Score for operator - new best solution	35
σ_2	Score for operator - new best current solution	2
σ_3	Score for operator - worse solution, but accepted via simulated annealing	1
n^{granular}	Initial number of nodes to be examined in granular insertion operator	30
min^{high}	Minimum acceptance threshold for insertion operator with high threshold	1.0
max^{high}	Maximum acceptance threshold for insertion operator with high threshold	10.0
min^{low}	Minimum acceptance threshold for insertion operator with low threshold	0.05
max^{low}	Maximum acceptance threshold for insertion operator with low threshold	0.5
ω	Empirical factor for prize calculation	2.0

(continued on next page)

Table B.6 (continued).

References

Albareda-Sambola, M., Fernández, E., & Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, *48*, 31–39. http://dx.doi.org/10.1016/j.cor.2014.02.010.

Amazon (2022). Amazon free no-rush shipping: How it works. URL: https://www.amazon.com/b?ie=UTF8&node=9433645011.

Angelelli, E., Bianchessi, N., Mansini, R., & Speranza, M. (2009a). Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C (Emerging Technologies)*, *17*(2), 106–119. http://dx.doi.org/10.1016/j.trc.2008.02.001.

Angelelli, E., Bianchessi, N., Mansini, R., & Speranza, M. G. (2009b). Management policies in a dynamic multi period routing problem. In *Innovations in distribution logistics* (pp. 1–15). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-540-92944-4_1.

Angelelli, E., Bianchessi, N., Mansini, R., & Speranza, M. G. (2010). Comparison of policies in dynamic routing problems. *Journal of the Operational Research Society*, *61*(4), 686–695. http://dx.doi.org/10.1057/jors.2009.17.

Angelelli, E., Grazia Speranza, M., & Savelsbergh, M. W. (2007). Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks*, *49*(4), 308–317. http://dx.doi.org/10.1002/net.20180.

Angelelli, E., Savelsbergh, M. W., & Grazia Speranza, M. (2007). Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters*, *35*(6), 713–721. http://dx.doi.org/10.1016/j.orl.2007.02.006.

Archetti, C., Jabali, O., & Speranza, M. G. (2015). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, *61*, 122–134. http://dx.doi.org/10.1016/j.cor.2015.03.014.

Bektaş, T., Repoussis, P. P., & Tarantilis, C. D. (2014). Chapter 11: Dynamic vehicle routing problems. In *Vehicle routing* (pp. 299–347). Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9781611973594.ch11.

Billing, C., Jaehn, F., & Wensing, T. (2018). A multiperiod auto-carrier transportation problem with probabilistic future demands. *Journal of Business Economics*, *88*(7–8), 1009–1028. http://dx.doi.org/10.1007/s11573-018-0904-x.

Buldeo Rai, H., Broekaert, C., Verlinde, S., & Macharis, C. (2021). Sharing is caring: How non-financial incentives drive sustainable e-commerce delivery. *Transportation Research Part D: Transport and Environment*, *93*, Article 102794. http://dx.doi.org/10.1016/j.trd.2021.102794.

Buldeo Rai, H., Verlinde, S., & Macharis, C. (2019). The "next day, free delivery" myth unravelled: Possibilities for sustainable last mile transport in an omnichannel environment. *International Journal of Retail & Distribution Management*, *47*(1), 39–54. http://dx.doi.org/10.1108/ijrdm-06-2018-0104.

Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, *63*(3), http://dx.doi.org/10.1002/net.21544, 276–276.

Cantu-Funes, R., Salazar-Aguilar, M. A., & Boyer, V. (2017). Multi-depot periodic vehicle routing problem with due dates and time windows. *Journal of the Operational Research Society*, *69*(2), 296–306. http://dx.doi.org/10.1057/s41274-017-0206-7.

Ceschia, S., Di Gaspero, L., & Schaerf, A. (2010). Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. *Journal of Scheduling*, *14*(6), 601–615. http://dx.doi.org/10.1007/s10951-010-0213-x.

Darvish, M., Archetti, C., Coelho, L. C., & Speranza, M. G. (2019). Flexible two-echelon location routing problem. *European Journal of Operational Research*, *277*(3), 1124–1136. http://dx.doi.org/10.1016/j.ejor.2019.04.002.

Darvish, M., Coelho, L. C., & Laporte, G. (2020). *The Dynamic Routing Problem with Due Dates and Stochastic Release Dates*. Faculté des sciences de l'administration.

Deloison, T., Hannon, E., Huber, A., Heid, B., Klink, C., Sahay, R., & Wolff, C. (2020). The future of the last-mile ecosystem. URL: https://www3.weforum.org/docs/WEF_Future_of_the_last_mile_ecosystem.pdf.

DHL (2021). Bücher- und Warensendung im Detail. URL: https://www.deutschepost.de/de/w/buecherundwarensendung.html.

Dietl, M., Voigt, S., & Kuhn, H. (2024). From rush to responsibility: Evaluating incentives on online fashion customers' willingness to wait. *Transportation Research Part D: Transport and Environment*, *133*, Article 104280. http://dx.doi.org/10.1016/j.trd.2024.104280.

Dragomir, A. G., Van Woensel, T., & Doerner, K. F. (2022). The pickup and delivery problem with alternative locations and overlapping time windows. *Computers & Operations Research*, *143*, Article 105758. http://dx.doi.org/10.1016/j.cor.2022.105758.

Dumez, D., Lehuédé, F., & Péton, O. (2021). A large neighborhood search approach to the vehicle routing problem with delivery options. *Transportation Research, Part B (Methodological)*, *144*, 103–132. http://dx.doi.org/10.1016/j.trb.2020.11.012.

Estrada-Moreno, A., Savelsbergh, M., Juan, A. A., & Panadero, J. (2019). Biased-randomized iterated local search for a multiperiod vehicle routing problem with price discounts for delivery flexibility. *International Transactions in Operational Research*, *26*(4), 1293–1314. http://dx.doi.org/10.1111/itor.12625.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, *39*(2), 188–205. http://dx.doi.org/10.1287/trsc.1030.0079.

Francis, P. M., Smilowitz, K. R., & Tzur, M. (2008). The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges* (pp. 73–102). Springer.

Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR. Information Systems and Operational Research*, *37*(3), 297–318. http://dx.doi.org/10.1080/03155986.1999.11732386.

Ignat, B., & Chankov, S. (2020). Do e-commerce customers change their preferred last-mile delivery based on its sustainability impact? *The International Journal of Logistics Management*, *31*(3), 521–548. http://dx.doi.org/10.1108/ijlm-11-2019-0305.

Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1989). Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research*, *37*(6), 865–892. http://dx.doi.org/10.1287/opre.37.6.865.

Keskin, M., Branke, J., Deineko, V., & Strauss, A. K. (2023). Dynamic multi-period vehicle routing with touting. *European Journal of Operational Research*, *310*(1), 168–184. http://dx.doi.org/10.1016/j.ejor.2023.02.037.

Kim, E. (2019). Amazon can already ship to 72% of US population within a day. URL: https://www.cnbc.com/2019/05/05/amazon-can-already-ship-to-72percent-of-us-population-in-a-day-map-shows.html.

Laganà, D., Laporte, G., & Vocaturo, F. (2021). A dynamic multi-period general routing problem arising in postal service and parcel delivery systems. *Computers & Operations Research*, *129*, Article 105195. http://dx.doi.org/10.1016/j.cor.2020.105195.

Larrain, H., Coelho, L. C., Archetti, C., & Speranza, M. G. (2019). Exact solution methods for the multi-period vehicle routing problem with due dates. *Computers & Operations Research*, *110*, 148–158. http://dx.doi.org/10.1016/j.cor.2019.05.026.

Maloni, M., Paul, J. A., & Gligor, D. M. (2013). Slow steaming impacts on ocean carriers and shippers. *Maritime Economics & Logistics*, *15*(2), 151–171. http://dx.doi.org/10.1057/mel.2013.2.

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. http://dx.doi.org/10.1016/j.trc.2015.06.016.

Muñoz-Villamizar, A., Velázquez-Martínez, J. C., Mejía-Argueta, C., & Gámez-Pérez, K. (2021). The impact of shipment consolidation strategies for green home delivery: a case study in a Mexican retail company. *International Journal of Production Research*, *60*(8), 2443–2460. http://dx.doi.org/10.1080/00207543.2021.1893852.

Nogueira, G. P. M., de Assis Rangel, J. J., & Shimoda, E. (2021). Sustainable last-mile distribution in B2C e-commerce: Do consumers really care? *Cleaner and Responsible Consumption*, *3*, Article 100021. http://dx.doi.org/10.1016/j.clrc.2021.100021.

Pacheco, J., Alvarez, A., García, I., & Angel-Bello, F. (2012). Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society*, *63*(5), 569–581. http://dx.doi.org/10.1057/jors.2011.51.

Pitney Bowes (2023). Pitney bowes parcel shipping index 2023. URL: https://www.pitneybowes.com/content/dam/pitneybowes/us/en/shipping-index/23-mktc-03596-2023_global_parcel_shipping_index_ebook-web.pdf,

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2015). Dynamic vehicle routing problems: Three decades and counting. *Networks*, *67*(1), 3–31. http://dx.doi.org/10.1002/net.21628.

Punakivi, M., & Saranen, J. (2001). Identifying the success factors in e-grocery home delivery. *International Journal of Retail & Distribution Management*, *29*(4), 156–163. http://dx.doi.org/10.1108/09590550110387953.

PyVRP (2023). PyVRP - a python library for vehicle routing problems. https://github.com/PyVRP/PyVRP.

Rave, A., & Fontaine, P. (2024). The location routing problem with time windows and load-dependent travel times for cargo bikes. *European Journal of Operational Research*, http://dx.doi.org/10.1016/j.ejor.2024.11.040.

Ritzinger, U., Puchinger, J., & Hartl, R. F. (2015). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, *54*(1), 215–231. http://dx.doi.org/10.1080/00207543.2015.1043403.

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. http://dx.doi.org/10.1287/trsc.1050.0135.

Soeffker, N., Ulmer, M. W., & Mattfeld, D. C. (2022). Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, *298*(3), 801–820. http://dx.doi.org/10.1016/j.ejor.2021.07.014.

Subramanyam, A., Mufalli, F., Laínez-Aguirre, J. M., Pinto, J. M., & Gounaris, C. E. (2021). Robust multiperiod vehicle routing under customer order uncertainty. *Operations Research*, *69*(1), 30–60. http://dx.doi.org/10.1287/opre.2020.2009.

Tilk, C., Olkis, K., & Irnich, S. (2021). The last-mile vehicle routing problem with delivery options. *OR Spectrum*, *43*(4), 877–904. http://dx.doi.org/10.1007/s00291-021-00633-0.

Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, *15*(4), 333–346. http://dx.doi.org/10.1287/ijoc.15.4.333.24890.

Ulmer, M. W. (2018). Horizontal combinations of online and offline approximate dynamic programming for stochastic dynamic vehicle routing. *Central European Journal of Operations Research*, *28*(1), 279–308. http://dx.doi.org/10.1007/s10100-018-0588-x.

Ulmer, M. W., Soeffker, N., & Mattfeld, D. C. (2018). Value function approximation for dynamic multi-period vehicle routing. *European Journal of Operational Research*, *269*(3), 883–899. http://dx.doi.org/10.1016/j.ejor.2018.02.038.

Vidal, T. (2022). Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, *140*, Article 105643. http://dx.doi.org/10.1016/j.cor.2021.105643.

Voigt, S. (2025). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research*, *322*(2), 357–375. http://dx.doi.org/10.1016/j.ejor.2024.05.033.

Voigt, S., Frank, M., Fontaine, P., & Kuhn, H. (2022). Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research*, *146*, Article 105856. http://dx.doi.org/10.1016/j.cor.2022.105856.

Voigt, S., Frank, M., Fontaine, P., & Kuhn, H. (2023). The vehicle routing problem with availability profiles. *Transportation Science*, *57*(2), 531–551. http://dx.doi.org/10.1287/trsc.2022.1182.

Wen, M., Cordeau, J.-F., Laporte, G., & Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, *37*(9), 1615–1623. http://dx.doi.org/10.1016/j.cor.2009.12.002.

Wiese, J. (2017). *Slow logistics: eine simulationsgestützte Analyse der ökonomischen und ökologischen Potentiale der Sendungsbündelung* (Dissertation), University of Bamberg Press.

Wouda, N. A., Aerts-Veenstra, M., & van Foreest, N. (2024). An integrated selection and routing policy for urban waste collection. URL: https://arxiv.org/abs/2409.13386. arXiv:2409.13386.

Wouda, N. A., Lan, L., & Kool, W. (2024). PyVRP: A high-performance VRP solver package. *INFORMS Journal on Computing*, http://dx.doi.org/10.1287/ijoc.2023.0055.

Yildiz, B., & Savelsbergh, M. (2020). Pricing for delivery time flexibility. *Transportation Research, Part B (Methodological)*, *133*, 230–256. http://dx.doi.org/10.1016/j.trb.2020.01.004.