OR Spectrum (2022) 44:795–832
https://doi.org/10.1007/s00291-021-00664-7

ORIGINAL ARTICLE

Integrated zone picking and vehicle routing operations with restricted intermediate storage

Manuel Ostermeier[1] · Andreas Holzapfel[2] · Heinrich Kuhn[3] · Daniel Schubert[3]

Received: 21 December 2020 / Accepted: 23 November 2021 / Published online: 29 December 2021

Abstract

The competitiveness of a retailer is highly dependent on an efficient distribution system. This is especially true for the supply of stores from distribution centers. Stores ask for high flexibility when it comes to their supply. This means that fast order processing is essential. Order processing affects different subsystems at the distribution center: orders are picked in multiple picking zones, transferred to intermediate storage, and delivered via dedicated tours. These processing steps are highly interdependent. The schedule for picking needs to be synchronized with the routing decisions to ensure availability of orders at the DC's loading docks when their associated tours are scheduled. Concurrently, intermediate storage represents a bottleneck as capacity for order storage is limited. The simultaneous planning of picking and routing operations with restricted intermediate storage is therefore relevant for retail practice but has not so far been considered within an integrated planning approach. Our work addresses this task and discusses an integrated zone picking and vehicle routing problem with restricted intermediate storage. We present a comprehensive model formulation and introduce a general variable neighborhood search for simultaneous consideration of the given planning stages. We also present two alternative sequential approaches that are motivated by the prevailing planning situation in industry. Numerical experiments and a case study show the need for an integrated planning approach to obtain practicable results. Further, we identify the impact of the main problem characteristics on overall planning and provide valuable insights for the application of these findings in industry.

Keywords Routing · Zone picking · Intermediate storage · General variable neighborhood search

✉ Manuel Ostermeier
manuel.ostermeier@tum.de

[1] Supply and Value Chain Management, Technical University of Munich, München, Germany

[2] Logistics Management, Geisenheim University, Geisenheim, Germany

[3] Supply Chain Management & Operations, Catholic University Eichstätt-Ingolstadt, Ingolstadt, Germany

1 Introduction

In grocery distribution, the majority of store demand is fulfilled from distribution centers (DCs). Retailers often operate their own distribution network to supply stores with products from different segments, i.e., different temperature zones. These segments have individual requirements for the delivery process (e.g., delivery of fresh products before store opening). Additionally, high flexibility is expected from the retailers, especially regarding the supply with fresh and dairy products, as stores are very frequently supplied with these segments and often order until late on the day before delivery. The time span between the receipt of an order and its supply is therefore small, less than 20 h in most cases.

This requires efficient order processing at DCs and hence a coordinated course of action for all DC sections involved. This comprises the picking of orders, the temporary storage of products in intermediate storage, and the actual delivery process. Each store order that arrives at the DC passes through these three subsequent stages. In practice, the planning is mainly driven by the routing decisions as the routing amounts to a substantial share of the cost and also determines the arrival time of deliveries at the stores. This means that the routing problem is solved first and the picking is only adapted to the given delivery schedule. However, this often results in non-feasible solutions as time restrictions are not met and the distribution process is delayed. A routing first, picking second strategy ignores the fact that picking and routing operations are highly dependent on each other and bound to tight time restrictions. The intermediate storage connects picking and routing operations but also constitutes a bottleneck for the processing of orders. Picking and routing decisions have to be aligned to achieve practicable—i.e., feasible—solutions. At the same time, the capacity of intermediate storage has to be respected.

This paper addresses the planning problem described by proposing an integrative approach. It is based on an actual application in retail industry. We introduce the consideration of zone picking and restricted intermediate storage and analyze the interdependence of the individual planning steps. While most publications in the literature focus on either the order picking problem (OPP) or the vehicle routing problem (VRP), there are few publications that consider routing and picking simultaneously and highlight the importance of integrated planning (see, e.g., Schmid et al. 2013; Schubert et al. 2018). Solving each process separately may result in optimal solutions for the distinct problems but does not necessarily offer a feasible solution for the complete planning problem, or may result in high overall logistics costs. Yet none of the existing publications deal with restricted intermediate storage or multiple picking zones. Both characteristics are, however, commonly found in the retail industry.

We therefore propose an integrated zone picking and vehicle routing problem with time windows and restricted intermediate storage (ZPRI_VRP) to enable joint consideration of all the subproblems involved. The problem formulated is solved with a specialized heuristic solution approach, namely a general variable neighborhood search (GVNS). In a case study and further numerical experiments,

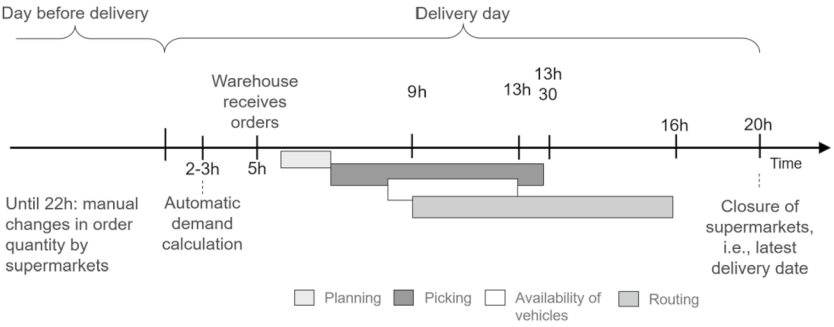


Fig. 1 Timeline for the distribution process

we show the effectiveness of our approach and the benefits of an integrated solution for picking and routing, assuming restricted intermediate storage. Moreover, we show the impact of distinct planning parameters and analyze under what conditions integrated planning is indispensable. The remainder of this paper is organized as follows. The problem motivated by retail industry is discussed in Sect. 2. We first introduce the processes involved before we describe their interrelations and the integrated planning problem. Existing literature related to our problem is discussed in Sect. 3. Section 4 describes a comprehensive model formulation for the ZPRI_VRP. The integrated solution approach developed and two alternative sequential approaches are presented in Sect. 5. Section 6 analyzes the benefits of integrated planning and further examines the impact of individual planning characteristics. Lastly, our findings are summarized in Sect. 7.

2 Problem

This section outlines the problem occurring in the retail industry. We briefly describe the overall setting before detailing the processes involved and their interrelations.

2.1 General setting and timeline

The DC at our case company is organized with respect to the given assortments for distribution. This means that there are dedicated warehouse sections for each product segment (e.g., ambient, dairy, fresh, and deep-frozen products), and each segment is distributed separately. Orders need to be prepared for distribution to the stores within each warehouse section. Despite the organization into different sections, this preparation process is similar across all sections and follows a defined chronology (as depicted in Fig. 1) that is common in retail practice.

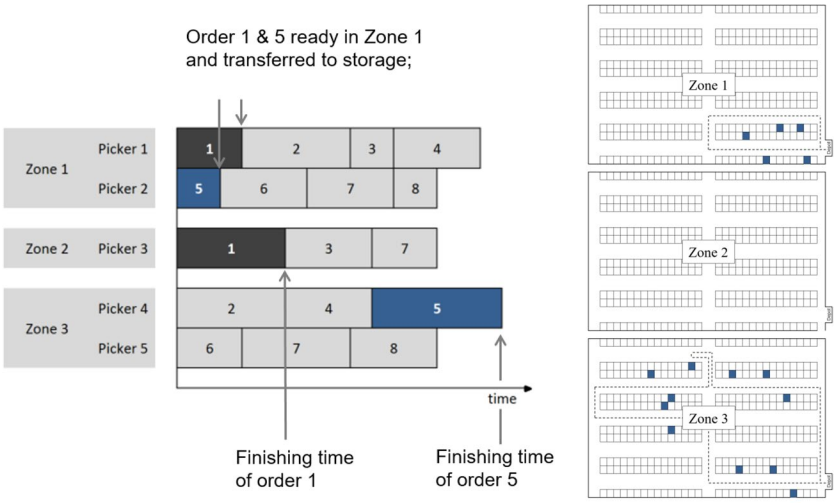
The general distribution process is valid for all segments and can be subdivided into three steps: (1) picking of orders, (2) storage of picked orders in intermediate storage, and (3) delivery of orders to stores. Typically, the precise order volumes are only available late the day before the delivery at the DC, as stores may change

orders at short notice. This is especially the case for frequently delivered product segments such as dairy products. After all orders are received, the planning process starts and the first orders are picked as soon as possible. The picking process lasts until all orders have been picked and prepared for transportation. The first tours start as soon as the corresponding orders are ready for their delivery. Our work focuses on the distribution of dairy products as currently executed at a major European grocery retailer. There the dairy product segment has the particularity that the time that vehicles are available for distribution is restricted. The restricted vehicle availability is due to the fact that the vehicles used for the distribution of dairy products are employed for the distribution of fresh products first. This means that the distribution fleet delivers fresh products first early in the morning, and vehicles return to the depot afterward for the delivery of dairy products. Both product segments require transportation in vehicles with cooling facilities, and therefore, vehicles are not exchanged with the fleets of other product segments.

2.2 Picking

The retailer's DC is designed to handle large order volumes and articles across various product segments. The picking areas for the distinct product segments are therefore divided into several zones, which take into account both product characteristics and warehouse sizes. Zones are defined for the different product categories and rotation speeds within the segment of dairy products for the setting at hand. The use of multiple zones has various advantages but also poses additional challenges for the planning process compared to traditional picking in single zones. So-called zone picking strategies are needed. Two strategies are basically applied to deal with zone picking (de Koster et al. 2007). In the *pick-and-pass* strategy, customer orders are passed from one zone to the next as soon as the picking process is completed in one zone. In contrast, partial orders may be picked independently of each other in different zones in *parallel zoning*. While the first strategy may prevent additional sorting or consolidation effort, parallel zoning may decrease the time span between the start of processing an order and its actual availability for delivery at the expense of additional complexity. In our problem setting, we consider a parallel zoning strategy. Consequently, a store order is divided into several suborders: one for each of the picking zones needed to pick all corresponding products. A suborder therefore defines a partial order that will be processed in one zone. Large order volumes, as are common in grocery retailing, imply large suborder volumes as well and thus suborders are picked discretely, i.e., an order picker processes one suborder at a time at most. This means each suborder is handled by one picker and no further split of (sub)orders is required.

Solving the zone picking problem (ZPP) determines a picking schedule for each zone. In detail, the suborders are assigned to the order pickers of the corresponding zone. The ZPP uses the processing time (pick time) of each order in the corresponding zone, which is determined by solving the picker routing problem. A solution for this problem can be obtained via simple routing strategies (see Roodbergen 2001) and is not part of our optimization problem. As the composition of each (sub)order



(a) Picking process across multiple zones (b) Picking for order 5

Fig. 2 Zone picking example

is known in advance, the picking times can be determined in a preprocessing step (Schubert et al. 2018). After a suborder has been picked, the order picker takes it to the intermediate storage area (see below). This point in time marks the finishing time of the corresponding suborder. A parent order is available for distribution as soon as all associated suborders have arrived in intermediate storage. In summary, the objective in our context is to define a picking sequence in which the suborders assigned are processed and become available for delivery tours.

Figure 2a provides an example solution of the ZPP for eight store orders. As can be seen, orders are subdivided into suborders and may be processed in parallel in all zones (see order 1 for example), or may not contain suborders in all zones (e.g., order 5). Figure 2b shows examples of picking routes for order 5 in the warehouse, which comprises items from zones 1 and 3, while no items from zone 2 are needed.

2.3 Intermediate storage

Intermediate storage represents the link between picking and routing operations within the distribution process. It is located between the order picking area of the DC and the loading docks. Its overall capacity is thus determined by the surface and the capacity measurements. Given the limitations of available storage space, the intermediate storage restricts the interaction of picking and routing by limiting the flow of orders between the two subproblems. The function of the intermediate storage is to (i) gather readily picked suborders, (ii) combine suborders of individual stores, (iii) sort store orders related to their associated delivery tours, and (iv) store orders until they are loaded for delivery. Figure 3 illustrates an example of the layout of intermediate storage within the DC, and the way it is modeled in our problem formulation.

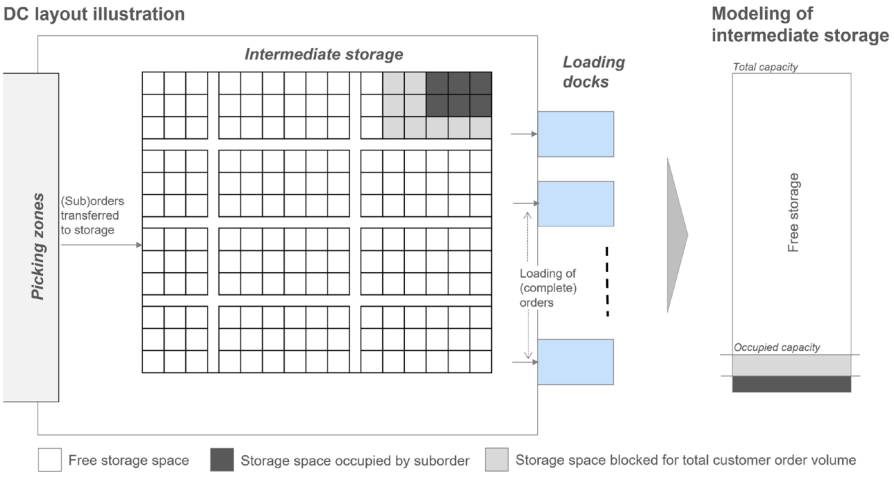


Fig. 3 Layout and modeling of intermediate storage (example)

Intermediate storage is organized as an independent storage area for all goods and is shared across all orders and tours, i.e., there is no dedicated space for each truck tour and corresponding loading dock, while all individual storage locations are accessible by the pickers. This enables picking operations to work on multiple tours (i.e., tours starting at different times and successively at the same docks) at the same time, and to provide orders of different tours simultaneously to the storage. The exact position of orders in the storage is not considered in our problem, but is reduced to the overall capacity consumption (see right-hand side of Fig. 3). The free space in the storage is time dependent due to the dynamics of picking and delivery: new (sub)orders enter the storage after picking, while complete customer orders leave the storage for delivery once all respective suborders are available and the tour is ready for loading. When a customer's first (sub)order starts its picking process (symbolized by the dark grey area in Fig. 3), the space for the total customer order volume is reserved (see light grey area in Fig. 3) to ensure enough capacity for remaining suborders such that a scheduled delivery tour can be executed as planned. This concept generally corresponds to the assumptions of a waveless picking system that involves the continuous transfer of individual orders, based on a priority ranking of store orders that respects target delivery time windows and tour planning, for example (Gallien and Weber 2010). However, the release strategy within the modeling approach proposed prevents the system from being blocked when a missing suborder cannot be moved from the picking area to the intermediate storage area due to lack of space. A limited number of loading docks is directly connected to the storage. The number of docks corresponds to the size of the intermediate storage as loading docks are established alongside the intermediate storage to enable direct access for loading.

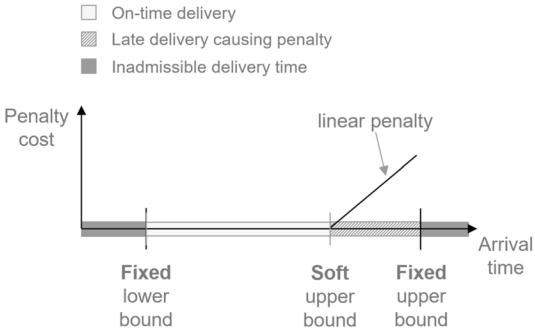


Fig. 4 Delivery times for the supply of stores

2.4 Vehicle routing

The delivery of dairy products from the DC to the stores is the last process in the line of distribution. It determines the tours for supplying the stores and thus the assignment of stores to tours and the sequence of visits. The delivery is done by homogeneous vehicles of a limited delivery fleet used both for fresh and dairy products. The availability of vehicles for the delivery of dairy products is restricted: Vehicles conduct one delivery tour for fresh products (not considered in our problem) early in the morning, after which they return to the depot and become available for the delivery of dairy products. Please note that each vehicle is used once at most per segment. A store order has to be fulfilled by a single delivery, and the delivery has to take place within an agreed-on time window. Generally, there are two possible types of delivery time window used in vehicle routing: hard and soft time windows. While hard time windows indicate fixed time bounds in which a delivery is possible (e.g., opening hours of stores), soft time windows indicate an agreed delivery time span that may be violated causing a penalty (soft bounds). In our case, we use a combination of both (see Fig. 4) that reflects a typical setting in retail practice and is applied at our partner retailer. It resembles a time window setting of type four as described in Fu et al. (2008). The arrival time at a store is restricted by a fixed lower bound. This means that a vehicle has to wait to begin its service if it arrives before that time. Equivalently, there is a fixed upper bound for the arrival at a store that cannot be violated. Additionally, retailer and store covenant a time by which a delivery has to happen, i.e., a soft upper bound. A penalty fee applies depending on the extent of the delay if the retailer fails to supply the store until the soft upper bound.

Subsequent to time windows, service times at customer locations for the unloading process have to be considered for the tour planning as well as loading times at the DC before the tour starts. The length of a tour is therefore determined by the loading time of a vehicle at the DC, the travel times between stores, and the corresponding service times at each location.

2.5 Integrated problem formulation

The integrated zone picking and vehicle routing problem with restricted intermediate storage considers the processes presented in Sects. 2.2 to 2.4. The individual process steps (picking, intermediate storage, and routing) are highly interrelated. To begin with, the order picking problem affects both intermediate storage and delivery as it determines at what time orders enter storage and are consequently available for vehicle routing. The arrival times of finished orders from picking impact the earliest possible start times for the delivery tours. The intermediate storage connects picking and delivery processes but also restricts both. The picking of (sub)orders may only start if enough capacity in intermediate storage is available. Otherwise, the picker is blocked and needs to postpone picking activities. In line with this, tours can only start if the orders are available. The restricted capacity of the intermediate storage therefore has to be used advisedly so as not to disconnect the picking and delivery problems. Lastly, the delivery process directly impacts picking decisions and storage. The loading of orders on tours frees capacity in storage and as such enables the picking of new orders. Vehicle routing seeks to find cost-optimal tours, while it is bound to tight time restrictions. Deliveries require punctual availability of orders to fulfill deliveries on time. Planned delivery tours and the corresponding start times of tours therefore dictate the required finishing times of orders and thus influence the picking schedule.

In summary, the processes involved need to be considered simultaneously and cannot be treated independently. We therefore propose an approach that simultaneously decides on (i) the order picking schedule and thus the corresponding finishing times of orders and (ii) the vehicle routing, which defines the tours, their start and the arrival times at store locations, while (iii) restricted intermediate storage capacity is considered. All three problem aspects impact the distribution planning as they define the time and setting for deliveries. We assess the impact of the individual processes on distribution costs by evaluating resulting routing, vehicle usage, and penalty costs.

3 Literature

This section discusses related literature. We detail the literature on order picking and vehicle routing as they constitute the subproblems relevant to our formulation. In addition, related problem constellations are discussed where two subsystems are connected by a restricted buffer area that must be taken into account when scheduling the subsystems. Finally, we review the publications on integrated order picking and vehicle routing problems.

3.1 Order picking problem

The order picking problem considered is characterized by parallel zone picking. The (sub)orders are picked discretely within each zone. Discrete order picking operations are not explicitly dealt with in literature as they are equivalent to machine scheduling problems, where order pickers represent machines and pick times represent machine-independent processing times (Scholz et al. 2017). Literature on corresponding problems has been reviewed by Potts and Strusevich (2009), Moons et al. (2017) and Schubert et al. (2018). These sources serve as a basis for several order picking applications and problems (see Sect. 3.4).

To the best of our knowledge, parallel zone picking operations have not yet been investigated as a separate problem at an operational level, e.g., regarding the determination of picking schedules across zones. However, adjacent problems have been combined with zone picking at a higher level. The review of van Gils et al. (2018b) presents a selection of these problems. They also show that zone picking operations have additionally been investigated in a few joint order picking systems. van Gils et al. (2018a) examine the benefits of integrative solution approaches to picking problems, such as order batching and zone picking. Alongside the benefits, the authors state that there is an increase in effort for related activities such as sorting.

3.2 Vehicle routing problem

There is a large body of literature on VRPs and their variants (e.g., capacitated VRPs, VRPs with time windows, see Toth and Vigo (2014)). We therefore focus on VRPs that relate to the main characteristics of our work. Consequently, we discuss two relevant streams: (i) VRPs with order release dates and (ii) VRPs with soft time windows.

(i) Order release dates. VRPs with order release dates consider order availability for routing and restrict the determination of tours and routes. However, release dates are used as input parameters in existing literature, while in our application these times are part of the decision problem. The complexity of such a problem structure has been investigated by Archetti et al. (2015a). Additionally, delivery due dates have been considered by Archetti et al. (2015b). Finally, Cattaruzza et al. (2016) propose a hybrid genetic algorithm for a VRP integrating multiple trips and order release dates.

(ii) Soft time window constraints. Fu et al. (2008) classify six different types of soft time window. Our problem setting belongs to type four, i.e., a hard lower bound and a soft upper bound that can be violated until the hard upper bound is reached. A problem setting with a hard time window constraint embracing a soft one has, for example, been studied by Mouthuy et al. (2015). In addition, the unified solution framework proposed by Vidal et al. (2014) can be used to solve VRPs with soft and hard time windows.

3.3 Restricted buffer areas

Restricted buffer areas between two interdependent subsystems can lead to blocking and starving situations. This applies to intermediate storage as considered in our application as well as to similar structures in other contexts. The first works regarding restricted buffer zones originate from chemical systems. Karimi and Reklaitis (1983) and Karimi and Reklaitis (1985), for example, investigate impacts and determine capacity of a storage tank between batch and semicontinuous processes. Scheduling literature usually considers intermediate storage as buffer between different production stages. For example, an application in the steel industry similar to a flow shop is considered by Witt and Voß (2007), who propose simple heuristics for scheduling with intermediate storage, and present a literature review to intermediate storage in scheduling.

In transportation, cross-docks can have intermediate storage or take over the role of intermediate storage, especially regarding the function of consolidating shipments. A large body of literature is available regarding cross-docking operations and planning problems such that we refer to van Belle et al. (2012) who provide a state-of-the-art review. Rijal et al. (2019), for example, consider the case where inbound and outbound trucks can be processed at the same dock doors, so called mixed-mode doors. Similar to our case, where picking and tour planning has to be planned simultaneously because of the limited storage space, an integrated planning approach is required when combining truck scheduling and mixed-mode dock-door assignment. Nevertheless, the available studies in cross-docking operations either deal with the dock door or temporary storage assignment such that both decisions have not been investigated simultaneously so far (van Belle et al. 2012).

Picker-to-part order picking with batching or zoning operations may include a sorting process after (sub)orders are retrieved (de Koster et al. 2007). This is also true for automated storage and retrieval systems or in a shuttle-based storage and retrieval system. A special sorter or central conveyor loop carries out the sorting process in several practical applications (de Koster et al. 2007). The sorting process then interconnects the picking operations with subsequent processes, but also restricts the system's performance due to limited capacities or throughput. Tappia et al. (2019) and Bansal and Roy (2021) investigate storage and retrieval operations in an order picking system that combines picking stations and an automated storage and retrieval system, i.e., a high bay warehouse, by a central conveyor. Both the picking stations and the rack system are restricted to the capacity of the central conveyor. Rijal et al. (2021) investigate workforce scheduling for distribution facilities. The authors determine the number of pickers necessary to operate a fixed workload, i.e., batches to be picked within certain time windows, and explicitly specify break and idle times in the operational schedule. However, none of the listed publications deal with a restricted buffer zone (intermediate storage) and connect picking and routing operations, as is the case in our application.

3.4 Integrated order picking and vehicle routing problems

The branch of research on integrated order picking and vehicle routing problems is relatively new. In contrast, integrated production and delivery problems have been extensively studied. Since the order picking process has features in common with (production) scheduling problems, corresponding literature on integrated problems is also of interest. The large body of associated literature has recently been surveyed by Chen (2010) and Moons et al. (2017), to which we refer for detailed insights. Schubert et al. (2018) further provide an overview on integrated machine scheduling and delivery problems. None of the problems presented is, however, similar to our setting.

Schmid et al. (2013) are the first to propose an integrated order picking and vehicle routing problem within a set of so-called rich routing problems. They propose a model formulation that deals with order batching decisions but neglects intermediate storage and loading dock restrictions. Subsequent to Schmid et al. (2013), further practical-orientated problems were dealt with. With the exception of Kuhn et al. (2021) who deal with order batching, all publications are based on discrete order picking operations with multiple pickers. Additionally, all contributions assume that vehicles are available right from the start of the planning horizon. This constitutes a major difference versus our modeling approach. Schubert et al. (2018) consider a limited number of order pickers and vehicles, though vehicles can perform multiple trips. They consider delivery due dates, i.e., they assume soft upper bounds but neglect lower bounds and evaluate total tardiness, while the following publications consider cost-objective functions. Schubert et al. (2021) investigate a same-day delivery scenario with vehicle-customer dependencies, i.e., a site-dependent VRP is considered. Both the number of pickers and vehicles per type to deploy have to be determined. In the problem dealt with in Moons et al. (2018), Moons et al. (2019) and Ramaekers et al. (2018), the number of order pickers and vehicles is fixed, but a limited number of additional pickers can be hired temporarily. The vehicle fleet is heterogeneous in terms of capacity. As stated above, Kuhn et al. (2021) deal with order batching in the DC. The number of available resources is fixed for both picking and delivery. The vehicle routing process is characterized by semi-soft time windows, i.e., a hard lower and a soft upper bound, aimed at minimizing total tardiness.

3.5 Summary and contribution

There are numerous publications on both order picking and routing problems, while less attention has been paid to the simultaneous consideration of both subproblems. The importance of aligning picking and routing operations has only been identified and studied more intensively in recent years. These publications highlight the need for the integrated planning of picking and routing operations, and show that distinct planning leads to suboptimal solutions both in terms of costs and feasibility. Table 1 summarizes the integrated order picking and vehicle routing contributions available in literature.

Table 1 Overview of publications dealing with integrated order picking and vehicle routing

Publication	Characteristics							
	Picking		Vehicle routing*			Interm.	Resource	Objective
	# zones	Strategy	TW	Fleet	# trips	storage	planning	(minimize)
Schubert et al. (2018)	Single	Discrete	No LB, soft UB	HO	Multi-ple	–	Fixed	Tardiness
Moons et al. (2018)	Single	Discrete	Hard LB & UB	HE/ HO	Single	–	Pickers	Cost
Ramaekers et al. (2018)	Single	Discrete	Hard LB & UB	HO	Single	–	Pickers	Cost
Moons et al. (2019)	Single	Discrete	Hard LB & UB	HO	Single	–	Pickers	Cost
Schubert et al. (2021)	Single	Discrete	Hard LB & UB	HO, SD	Single	–	Pickers & vehicles	Cost
Kuhn et al. (2021)	Single	Batching	Hard LB, soft UB	HO	Single	–	Fixed	Tardiness
This research	Multi-ple	Discrete, parallel zoning	Hard LB, soft & hard UB	HO, restricted availabil-ity	Single	√	Fixed	Cost

**TW* time window; *LB* lower bound; *UB* upper bound; *HO* homogeneous; *HE* heterogeneous; *SD* vehicle-site dependencies

Table 1 also reveals the novelty of our modeling approach compared to existing integrated order picking and vehicle routing approaches published so far. Ours mainly differs from existing approaches in the following respects. (i) It considers a zone picking process; (ii) it assumes hard lower and soft upper bounds, but restricts the possible delivery times by additional hard upper bounds; (iii) it assumes that vehicle availability for delivery is restricted; and, most importantly, (iv) it considers a limited intermediate storage area between picking and delivery. All these additional aspects are common in retail practice. We therefore substantially contribute to the available body of literature on integrated order picking and routing problems. We present an innovative approach that aligns picking and routing processes and additionally takes into account restricted intermediate storage, vehicle availability, and multiple picking zones. The consideration of intermediate storage (via modeling time-dependent inventory and capacity restrictions), restricted vehicle availability (adding restrictions on tour loading and start times), and multiple zones (resulting in partial orders that need to be considered simultaneously within picking schedules and for order availability) lead to major changes both in terms of the practical problem and the resulting model formulation.

4 Mathematical model

This section presents the formal model formulation of the ZPRI_VRP. An overview of the sets and parameters involved is presented in Table 2.

Table 2 Sets and parameters for the ZPRI_VRP

Sets	
N	Set of stores, $N = \{1, \dots, n\}$, $N^0 = N \cup \{0\}$ with 0 as DC
N_z	Set of stores with a suborder to be processed in zone $z, z \in Z$; $N_z \subseteq N$
Z	Set of picking zones, $Z = \{1, \dots, \lvert Z \rvert\}$
Z_i	Set of zones in which a suborder of store $i, i \in N$ has to be processed, $Z_i \subseteq Z$
P_z	Set of order pickers available in zone $z, z \in Z$
T	Set of time periods, $T = \{1, \dots, \lvert T \rvert\}$
K	Set of vehicles, $K = \{1, \dots, \lvert K \rvert\}$
L	Set of loading docks, $L = \{1, \dots, \lvert L \rvert\}$
Parameters	
$\underline{a}_i, \overline{a}_i$	Hard lower/upper bound for arrival at store $i, i \in N$
$\overline{b}_i$	Soft upper bound for arrival at store $i, i \in N$
q_i	Volume of order $i, i \in N$
Q	Vehicle capacity
$\hat{Q}$	Intermediate storage capacity
d_{ij}	Costs for traveling from location i to $j, i, j \in N^0$
p_{iz}	Picking time for the suborder of store $i, i \in N_z$ in zone $z, z \in Z$
g_i	Total loading time for the order of store $i, i \in N$
M	Sufficiently large number
α_k	Availability time of vehicle $k, k \in K$
τ_{ij}	Travel time between locations i and $j, i, j \in N^0$
ψ_i	Service time at store $i, i \in N$
$\tilde{\omega}$	Cost parameter for vehicle usage per time unit
$\overline{\omega}_i$	Cost parameter for tardiness per store $i, i \in N$, per time unit

Overall structure. The ZPRI_VRP is defined on a complete undirected, weighted graph $G = (N^0, E)$, where $N^0 = \{0, 1, \dots, n\}$ is the set of nodes, including the set of stores $N = \{1, \dots, n\}$ and the depot ($\{0\}$). $E = \{(i,j) : i, j \in N^0\}$ is the set of edges. Each edge $(i,j) \in E$ is attributed to a given travel cost d_{ij} and time τ_{ij}. Additionally, the supply of each store requires a service time ψ_i. The planning horizon is defined by the discrete time set T. All actions are performed to the end of each period, e.g., if the finishing time of a (sub)order is $t, t \in T$, the (sub)order is available in the intermediate storage at $t + 1$.

Picking and intermediate storage. We consider multiple picking zones $z, z \in Z$. A defined set of pickers P_z is available in each zone, and the different zones in which a suborder of store i must be picked are indicated by the subset $Z_i, Z_i \subseteq Z$. The set of stores with a suborder that needs to be picked in zone z is further indicated by the subset $N_z, N_z \subseteq N$. The picking is carried out in each zone, and hence, a defined picking time p_{iz} is given for each suborder of a store i in zone z. After picking, the suborders are brought to the intermediate storage that has a limited capacity $\hat{Q}$ and is indicated in transportation units (TUs, e.g., roll-cages). TU is used for the capacity metric in order picking as well as for the tour capacities. (Sub)orders are indicated in full TUs, and

Table 3 Decision variables for the ZPRI_VRP

Decision variables	
u_{ik}	Binary variable indicating whether location $i, i \in N^0$, is assigned to vehicle $k, k \in K$, $(u_{ik} = 1)$ or not $(u_{ik} = 0)$
χ_{ijk}	Binary variable indicating whether location $i, i \in N^0$, is immediately approached before location $j, j \in N^0$, by vehicle $k, k \in K$
y_{iz}	Binary variable indicating whether the order of store $i, i \in N_z$, is processed first by an order picker $(y_{iz} = 1)$ or not $(y_{iz} = 0)$ in zone $z, z \in Z$
z_{ijz}	Binary variable indicating whether the order of store $i, i \in N_z$, is processed immediately before the order of store j, $j \in N_z, (z_{ijz} = 1)$ or not $(z_{ijz} = 0)$ in zone $z, z \in Z$
λ_{ik}	Binary variable indicating whether the order of store $i, i \in N$, is loaded first on vehicle k, $k \in K$
π_{ijk}	Binary variable indicating whether the order of store $i, i \in N$, is loaded immediately before the order of store j, $j \in N$, on vehicle k, $k \in K$
γ_k	Binary variable indicating whether vehicle $k, k \in K$, is assigned as the first vehicle to be handled at one of the available docks, i.e., if vehicle k is first in sequence at a dock
δ_{mk}	Binary variable indicating whether vehicle m is assigned immediately before vehicle k to a loading dock, $k, m \in K$
Auxiliary variables	
e_i	Actual time when the order of store $i, i \in N$, is completely loaded
o_k	Total operating time of vehicle $k, k \in K$
r_{iz}	Finishing time of the suborder of store $i, i \in N_z$, in zone $z, z \in Z$
$\underline{r}_i$	Finishing time of the first suborder of store $i, i \in N$, across all zones
$\overline{r}_i$	Finishing time of the complete order of store $i, i \in N$, i.e., latest finishing time across all corresponding suborders
s_k	Actual start time of vehicle $k, k \in K$, at DC
$\underline{x}_{it}$	Binary variable indicating whether at least one suborder of store $i, i \in N$, is or was located in the intermediate storage in time $t, t \in T$
$\overline{x}_{it}$	Binary variable indicating whether the order of store $i, i \in N$, is or was loaded completely to a vehicle in time $t, t \in T$
ϕ_i	Actual arrival time at store $i, i \in N$
$\overline{\phi}_i$	Tardiness at store $i, i \in N$

therefore, no consolidation is performed in the intermediate storage. Storage capacity is freed as soon as the order is completely loaded.

Loading and vehicle routing. A homogeneous fleet of vehicles K with capacity Q are available for distribution. The loading of a vehicle can start at the earliest when the corresponding vehicle is available at the depot, indicated by $\alpha_k, k \in K$. However, an associated set of limited loading docks L is available corresponding to the limited intermediate storage. The limited availability of loading docks may thus result in additional restrictions for the start times of the vehicles. Moreover, only complete orders can be loaded into vehicles as orders cannot be split. The loading time of store order i is given by g_i. Possible delivery times at each store $i, i \in N$, are limited by hard time windows $[\underline{a}_i, \overline{a}_i]$. Further, an additional time window determined by $[\underline{a}_i, \overline{b}_i]$ defines the contracted delivery times with a soft upper bound, where $\overline{b}_i \leq \overline{a}_i$. This means that the upper bound $\overline{b}_i$ may

be exceeded but causes additional store-specific costs. The costs of a vehicle are denoted by $\tilde{\omega}$, indicating the vehicle usage costs per time unit. Lastly, in case a vehicle arrives late at a store, the store-specific cost parameter $\overline{\omega}_i$ indicates the tardiness penalty per time unit. It represents service agreements that may differ as it is individual per store.

Mathematical model. For the mathematical model, we introduce the decision and auxiliary variables given in Table 3. Using the defined sets, parameters and decision variables, the ZPRI_VRP can be formulated as follows.

$$\textit{Minimize} \quad \sum_{k\in K}\sum_{i\in N^0}\sum_{j\in N^0} d_{ij}\cdot\chi_{ijk} + \tilde{\omega}\sum_{k\in K} o_k + \sum_{i\in N}\overline{\omega}_i\cdot\overline{\phi}_i \tag{1}$$

subject to

$$\sum_{i\in N_z} y_{iz} \le |P_z| \quad \forall z\in Z \tag{2}$$

$$y_{iz} + \sum_{j\in N_z, i\neq j} z_{jiz} = 1 \quad \forall z\in Z, i\in N_z \tag{3}$$

$$\sum_{\substack{j\in N_z\cup\{n+1\}\\ i\neq j}} z_{ijz} = 1 \quad \forall z\in Z, i\in N_z \tag{4}$$

$$r_{iz} \ge p_{iz} \quad \forall z\in Z, i\in N_z \tag{5}$$

$$r_{iz} \ge r_{jz} + p_{iz} - M\left(1 - z_{jiz}\right) \quad \forall z\in Z, i,j\in N_z, i\neq j \tag{6}$$

$$\underline{r}_i \le r_{iz} \quad \forall i\in N, z\in Z_i \tag{7}$$

$$\overline{r}_i \ge r_{iz} \quad \forall i\in N, z\in Z_i \tag{8}$$

$$\sum_{\underline{r}_i < t < |T|} \underline{x}_{it} = |T| - \underline{r}_i \quad \forall i\in N \tag{9}$$

$$\sum_{1\le t\le \underline{r}_i} \underline{x}_{it} = 0 \quad \forall i\in N \tag{10}$$

$$\sum_{e_i\le t<|T|} \overline{x}_{it} \ge |T| - e_i \quad \forall i\in N \tag{11}$$

$$\sum_{1\le t<e_i} \overline{x}_{it} = 0 \quad \forall i\in N \tag{12}$$

$$\sum_{i \in N} \left(\underline{x}_{it} - \overline{x}_{it} \right) \cdot q_i \leq \hat{Q} \quad \forall t \in T \tag{13}$$

$$\sum_{k \in K} \gamma_k = |L| \tag{14}$$

$$\gamma_k + \sum_{\substack{m \in K \\ m \neq k}} \delta_{mk} = 1 \quad \forall k \in K \tag{15}$$

$$\sum_{\substack{m \in K \cup \{|K|+1\} \\ m \neq k}} \delta_{mk} = 1 \quad \forall k \in K \tag{16}$$

$$\sum_{i \in N} \lambda_{ik} = 1 \quad \forall k \in K \tag{17}$$

$$\lambda_{ik} + \sum_{\substack{j \in N \\ i \neq j}} \pi_{ijk} = 1 \quad \forall k \in K, i \in N \tag{18}$$

$$\sum_{\substack{j \in N \cup \{n+1\} \\ i \neq j}} \sum_{k \in K} \pi_{ijk} = 1 \quad \forall i \in N \tag{19}$$

$$\pi_{ijk} \leq u_{ik} \quad \forall i, j \in N : i \neq j,\ k \in K \tag{20}$$

$$\lambda_{ik} \leq u_{ik} \quad \forall i \in N,\ k \in K \tag{21}$$

$$e_i \geq \overline{r_i} + g_i \quad \forall i \in N \tag{22}$$

$$e_i \geq s_m + g_i - M(2 - \delta_{mk} - u_{ik}) \quad \forall i \in N,\ k, m \in K \tag{23}$$

$$e_i \geq \alpha_k + g_i - M(1 - u_{ik}) \quad \forall i \in N,\ k \in K \tag{24}$$

$$s_k \geq e_i - M(1 - u_{ik}) \quad \forall k \in K, i \in N \tag{25}$$

$$\sum_{k \in K} u_{ik} = 1 \quad \forall i \in N \tag{26}$$

$$\sum_{k \in K} u_{0k} = |K| \tag{27}$$

$$\sum_{i \in N^0} \chi_{ijk} = u_{jk} \quad \forall j \in N, k \in K \tag{28}$$

$$\sum_{j \in N^0} \chi_{ijk} = u_{ik} \quad \forall i \in N, k \in K \tag{29}$$

$$\sum_{j \in N} \chi_{0jk} \leq 1 \quad \forall k \in K \tag{30}$$

$$\sum_{i \in N} u_{ik} \cdot q_i \leq Q \quad \forall k \in K \tag{31}$$

$$\phi_j \geq s_k + \tau_{0j} - M \cdot \left(1 - \chi_{0jk}\right) \quad \forall j \in N, k \in K \tag{32}$$

$$\phi_j \geq \phi_i + \psi_i + \tau_{ij} - M \cdot \left(1 - \chi_{ijk}\right) \quad \forall i, j \in N, i \neq j, k \in K \tag{33}$$

$$\underline{a_i} \leq \phi_i \leq \overline{a_i} \quad \forall i \in N \tag{34}$$

$$\overline{\phi}_i \geq \phi_i - \overline{b}_i \quad \forall i \in N \tag{35}$$

$$o_k \geq \phi_i - \alpha_k + \psi_i + \tau_{i0} - M \cdot (1 - u_{ik}) \quad \forall i \in N, k \in K \tag{36}$$

$$o_k, s_k \geq 0 \quad \forall k \in K \tag{37}$$

$$r_{iz} \geq 0 \quad \forall z \in Z, i \in N_z \tag{38}$$

$$e_i, \underline{r}_i, \overline{r}_i, \phi_i, \underline{\phi}_i, \overline{\phi}_i \geq 0 \quad \forall i \in N \tag{39}$$

$$u_{ik} \in \{0, 1\} \quad \forall i \in N^0, k \in K \tag{40}$$

$$\underline{x}_{it}, \overline{x}_{it} \in \{0, 1\} \quad \forall i \in N, t \in T \tag{41}$$

$$y_{iz} \in \{0, 1\} \quad \forall z \in Z, i \in N_z \tag{42}$$

$$z_{ijz} \in \{0, 1\} \quad \forall z \in Z, i \in N_z, j \in N_z \cup \{n + 1\} \tag{43}$$

$$\chi_{ijk} \in \{0,1\} \quad \forall i,j \in N^0, k \in K \tag{44}$$

The objective function (1) minimizes the total costs for driving, vehicle operation times and tardiness for deliveries. Note that all components of the objective function are evaluated in cost terms, either directly (traveling costs d_{ij}) or using the cost parameters introduced. The model constraints can be divided into three subsets: Constraints (2)–(13) define the order picking problem and the restricted intermediate storage. Loading and vehicle routing are described by Constraints (14)–(36), while (37)–(44) define the variable domains.

Picking and intermediate storage. The number of first orders in a picking sequence of a picker is limited by the number of pickers available in each zone (2). Consequently, a store order is either processed first or is the direct successor of another order in the picking sequence (3). The correct picking sequence is further guaranteed by (4). The finishing time of a suborder is at least as long as its picking time if it is first in sequence (5), or it is the sum of picking time and the finishing time of its direct predecessor in the picking sequence (6). The finishing times of the first and last suborders of store i determine the presence of the parent order ($\underline{r_i}, \overline{r_i}$) in the intermediate storage as given in (7) and (8). Using the arrival time of the first suborder ($\underline{r_i}$), the required capacity of the parent order is blocked in the intermediate storage (9), while the order is not assigned to the storage as long as it is not released (10). Constraints (11) and (12) determine the period when the loading of a store order is completed and the corresponding capacity is freed in the intermediate storage. Constraint (13) ensures that the available capacity of the intermediate storage is not exceeded. Please note that due to (7), (10) and (13), an order is picked as soon as enough capacity in the intermediate storage is available at the end of the picking process. For instance, (sub)order i is picked in zone z at time t if enough capacity is available at time $t + p_{iz}$. Otherwise, the corresponding picker is blocked until enough space is freed again.

Loading and vehicle routing. Similar to the picking sequence, vehicles need to be sequenced at the loading docks ((14)–(16)) and for the loading processes ((17)–(21)). The number of docks is limited, and hence, a limited number of vehicles can be assigned as the first vehicle to a dock (14). Further, each vehicle is either first in the sequence or a direct successor (15), while the correct sequence has to be respected (16). Only one store order can be first in the loading sequence of a vehicle (17). The loading sequence is then controlled by Constraints (18) and (19). Additionally, the loading and corresponding sequence is only relevant if an order is actually assigned to a vehicle ((20) and (21)). The time when the loading of an order is completed is defined by Constraints (22)–(24). The loading time of an order must be greater than (i) the finishing time from picking plus loading time (22), (ii) the departure of the preceding vehicle plus loading time in the event that it is the first order to be loaded (23), and (iii) the vehicle availability plus loading time in the event that the corresponding vehicle is the first in a sequence (24). Constraint (25) ensures that a vehicle only departs after all assigned orders are completely loaded. Each order has to be assigned to exactly one vehicle (26), and every vehicle has to depart from the depot ((27) and (30)). Constraints (28) and (29) ensure that each store is visited only once and that if a store is approached,

it has to be left again. The capacity restriction of vehicles is denoted by (31). The arrival times at stores are determined by (32) and (33) for the first store in the sequence and the remaining ones, respectively. Adherence to the hard time windows defined is ensured by (34). Tardy deliveries within the range allowed are determined by Constraint (35). The total operating time of vehicles is defined by Constraint (36).

5 Solution approach

Our solution approach simultaneously addresses the vehicle routing and picking problem, while respecting the restrictions for intermediate storage. Additionally, we present two alternative solution approaches that solve the ZPRI_VRP sequentially, as it is performed at our partner in industry. This allows us to analyze and evaluate different approaches, especially with respect to the common planning situation in practice. In the following, we first detail our integrated solution approach in Sect. 5.1, before discussing the two alternatives in Sect. 5.2.

5.1 Integrated general variable neighborhood search approach

The integrated solution approach is based on the general variable neighborhood search (GVNS) metaheuristic. The GVNS provides high-quality solutions in similar problem settings by approaching the interrelations of picking and routing operations in an efficient and effective manner (see, e.g., Schubert et al. 2021). Moreover, basic variable neighborhood search (VNS) formulations are used as state-of-the-art approaches for different vehicle routing (see, e.g., Pirkwieser and Raidl 2008; Hemmelmayr et al. 2009; Kovacs et al. 2014; Salhi et al. 2014) and order picking (see Henn (2015)) problems. We follow the GVNS and VND framework and refer to Hansen and Mladenović (2001) and Schubert et al. (2021) for a more detailed description. We propose a tailored solution approach based on this framework, the integrated General Variable Neighborhood Search (iGVNS), that tackles both routing and picking operations, while taking into account given intermediate storage restrictions. Figure 5 illustrates the framework of the integrated approach.

After an initial solution is found (Sect. 5.1.1), the improvement phase of the iGVNS starts. The improvement phase consists of a shaking phase (Sect. 5.1.2) and a subsequent local search, the VND (Sect. 5.1.3). The intermediate storage restricts both picking and routing decisions, and consequently, any new solution that is found has to be evaluated with respect to the storage restrictions. To do so we deploy a control logic for the intermediate storage (Sect. 5.1.4). Finally, a new solution is accepted according to defined criteria, and the search stops when the termination criterion is met (Sect. 5.1.5).

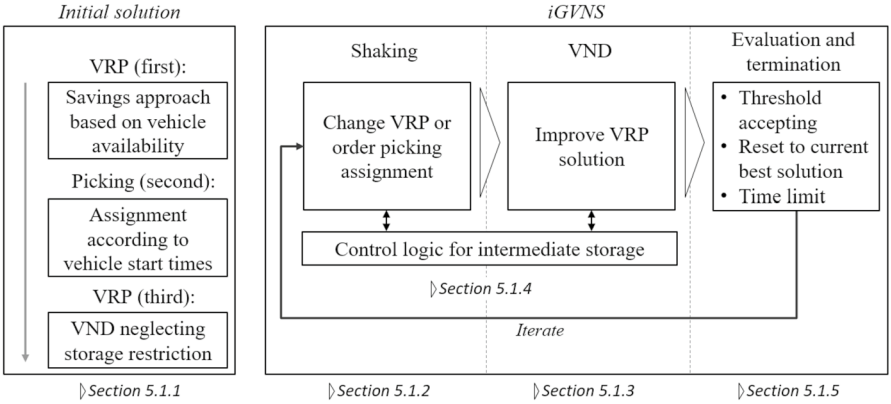


Fig. 5 Framework of the integrated solution approach

5.1.1 Initial solution

Customers are assigned to vehicles by a saving procedure (see Clarke and Wright (1964)) for an initial routing solution, neglecting the intermediate storage and picking operations, i.e., all vehicles used start as soon as they are available at the DC. Afterward, orders are assigned to pickers according to the potential start times of the assigned vehicles determined by the routing decisions of the savings procedure. Orders assigned to the same vehicle are prioritized by the largest processing time rule. Finally, the VND is used (see Sect. 5.1.3) to align the routing to the picking times determined in order to ensure reasonable starting times and delivery routes. Please note that this procedure neither guarantees a feasible solution with respect to the hard time window constraints nor regarding intermediate storage capacity. However, it provides a starting solution for coordinated picking and routing operations.

5.1.2 Shaking phase of iGVNS

The shaking phase is responsible for systematically changing the incumbent solution by exploiting different neighborhood structures. Our approach uses five types of neighborhood structure in the shaking phase. As we consider routing and picking decisions simultaneously, both decisions are considered within the neighborhoods. The following five neighborhoods are used.

1. Swap tours (*swap_VRP*): Interchanges two tours between the vehicles assigned. Vehicles may be available at different times or operated successively at the same loading dock. This means the start times of the two tours are affected as well as the start times of tours processed successively at the corresponding ramp(s), since loading times of the chosen tours may also vary.

2. Cross-exchange (*cross_VRP*): Swaps the sequences of stores of two different vehicles/tours (Taillard et al. 1997). Note that the sequence lengths may be different and one sequence may be empty, resulting in a reduction of one tour and an extension of the other tour.
3. Inverse cross-exchange (*icross_VRP*): Swaps sequences as the cross_VRP operator, but with an inversion of sequences.
4. Shift orders (*shift_OP*): Shifts an order to another picker in the same zone (Henn 2015; Scholz et al. 2017).
5. Swap orders (*swap_OP*): Swaps two sequences of orders between two order pickers of the same zone (Henn 2015; Scholz et al. 2017).

In the shaking phase, one neighborhood structure is chosen randomly among the available neighborhoods and the corresponding setting. Shaking is performed for the current incumbent solution that results from the local search (see Sect. 5.1.3). The structures employed and the parameter settings used for each neighborhood are summarized in Table 4.

5.1.3 Variable neighborhood descent approach

The local search procedure is applied after every shaking and exploits the current neighborhood until a new local optimum is found. It is therefore responsible for the extensive computing time of the iGVNS. As a consequence, the neighborhoods considered have to be kept small to enable efficient use of the VND within our search (Hansen et al. 2010). During the search, the neighborhoods defined are explored deterministically, i.e., in a predefined sequence. We integrate the following two types of neighborhood structure within the VND.

- Relocate customer (*relocate_VRP*): Shifts a store or a sequence of up to three stores to another position on the same route.
- Cross-exchange (*cross_X*): Swaps sequences of stores of two different vehilcles (tours) (Taillard et al. (1997)). It is performed only for small sequence lengths.

Table 4 Shaking structures and parameter specification

Corresponding problem	Neighborhood structure	Minimum	Maximum
		Number of stores	
	cross_VRP	0	3
Routing	*cross_VRP*	0	5
	icross_VRP	0	3
	icross_VRP	0	5
		Number of orders	
Picking	*shift_OP*	1	4
	swap_OP	1	5

The neighborhood structures are explored in the sequence and parameter setting as given in Table 5. The local search together with the shaking phase represent the improvement phase of the iGVNS. The algorithmic structure of the complete improvement phase is given in Appendix A.

5.1.4 Intermediate storage and control logic

The changes applied in the improvement phase directly impact the capacity usage in the intermediate storage. As we consider restricted intermediate storage, the capacity restriction has to be taken into account when changing the routing and/or picking.

To ensure adherence to the intermediate storage restriction (i.e., the capacity restriction) while preventing blocking situations, we propose a control logic that is depicted in Figure 6. This logic prioritizes resolving the order picking problem while retaining the current delivery schedule as far as possible, as delivery is the only cost-relevant operation (see Equation (1)). The control logic is applied to a current (promising) solution that has so far ignored the size of the intermediate storage. This means that picking and routing operations are aligned and, as order pickers are not exposed to idle times, all vehicles start at their earliest possible departure times.

5.1.5 Acceptance, termination and search space

The acceptance of new solutions follows two criteria. On the one hand, threshold accepting (see Dueck and Scheuer 1990; Henke et al. 2015) is used for diversification, while, on the other hand, the incumbent solution is reset to the best-known solution after a certain number of iterations without improvement. Further, the search terminates after a predefined computing time.

The ZPRI_VRP includes soft and hard time window bounds. While a violation of the soft upper bound only impacts the objective value of the current solution S' ($f(S')$, see (1)), a violation of hard time windows results in non-feasible solutions. We allow the violation of hard time windows within our search but steadily increase the corresponding penalty term ζ (see, e.g., Vidal et al. (2012)) to guide the search for feasible solutions. The initial penalization parameter ζ is updated after a certain number of

Table 5 VND structures: Parameter specification

Sequence	Neighborhood structure	Number of stores	
1	*relocate_VRP*	1	
2	*relocate_VRP*	2	
3	*relocate_VRP*	3	
		Minimum	Maximum
4	*cross_X*	1	1
5	*cross_X*	1	2
6	*cross_X*	1	3

iterations. We apply ζ to obtain the following artificial objective function $f_a(S')$, which is used during the search.

$$f_a(S') = f(S') + \zeta \cdot \sum_{i \in N} tardiness_i \tag{45}$$

Here, $tardiness_i$ quantifies the inadmissible early/late arrival at customer $i, i \in N$. Using this objective function, we enable a less restricted search and do not exclude promising solutions.

5.2 Alternative approaches for the ZPRI_VRP

In addition to the iGVNS, we present two alternative solution approaches. Both address the routing and picking problem sequentially and not in an integrated way.

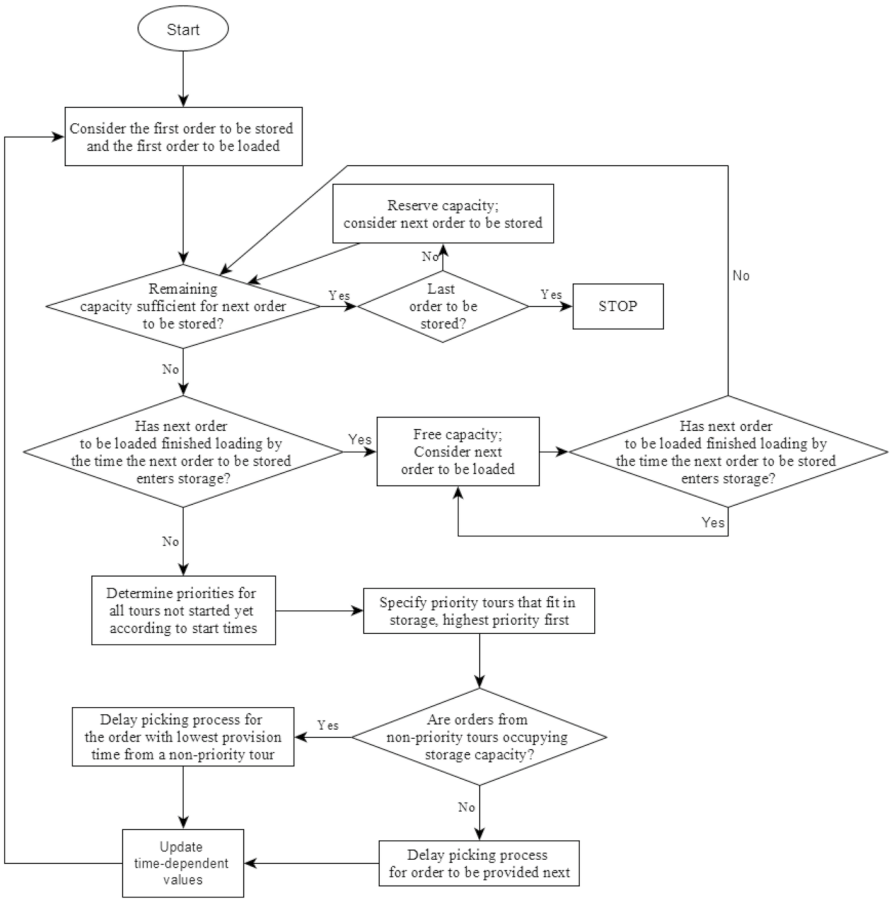


Fig. 6 Control logic for intermediate storage

Practically orientated solution approach. The first alternative solution approach, denoted as Seq_Prac, is inspired by the actual planning situation at our partner company in retail practice. Order picking and delivery planning are usually assigned to different divisions at companies. The delivery planning directly impacts service level at the stores and transportation costs. Order picking, on the other hand, has to ensure favorable delivery plans. The decision-making process therefore results in a routing first, picking second strategy. We adopt this approach and solve the routing problem first. This means that we first determine an initial routing solution (see Sect. 5.1.1) and improve this solution using the GVNS framework presented in Sect. 5.1, excluding picking operators. In this way, we first determine a final routing solution that is then used to align picking operations in a second step. The picking problem is therefore aligned based on the determined start times of tours (see Sect. 5.1.1). The intermediate storage restriction is afterward considered in the last step. The grey-colored area (Seq_Prac) in Figure 7 shows the steps of the solution approach typically applied in retail practice.

Iterated-sequential solution approach. The second alternative approach, denoted as Seq_Iter, extends the practical solution approach (Seq_Prac) by sequentially solving the routing and picking problem in an iterative procedure. In line with this, the GVNS framework is again used to improve the routing solution. More precisely, the GVNS framework is applied without the shaking operators for picking (see Seq_Prac) for a predefined number of iterations and the picking solution is aligned. Finally, the intermediate storage is considered using the control logic presented (see Sect. 5.1.4) and a new solution is accepted according to the criteria defined (see Sect. 5.1.5). Figure 7 illustrates the iterated-sequential solution approach.

6 Numerical results

This section presents experiments to assess the impact of intermediate storage and the integrated planning of routing and picking. Section 6.1 presents the data used and its generation process. Section 6.2 then analyses the solution quality of the

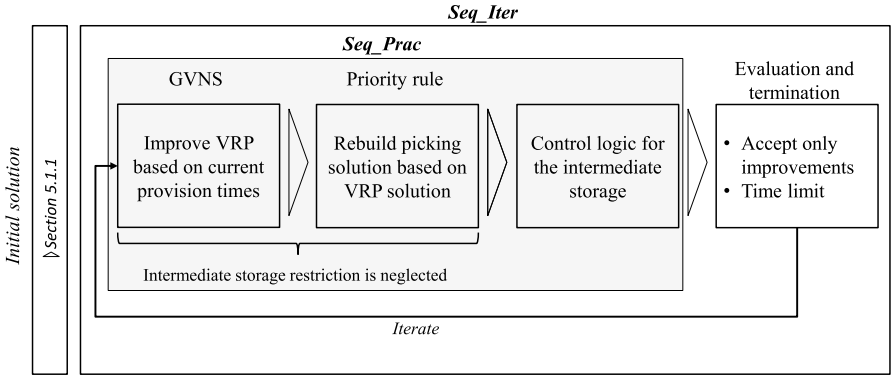


Fig. 7 Framework of the sequential approaches

iGVNS compared to an exact approach. Afterward, we present a case study to compare the iGVNS proposed to the approach used in retail practice (Sect. 6.3). We further provide an extended comparison for randomly generated instances and analyze problem specifics and their impact on the planning problem (Sect. 6.4).

6.1 Data generation and problem classes

General data. The data presented in the following are generally used for all experiments conducted. In the event of changes to the setting, we indicate the respective alternative setting in the corresponding section. We use the insights from our industry collaboration in the data generation process. In detail, we derive the basic cost structure, the characteristics of the vehicles, markets (e.g., order parameters), and the warehouse (e.g., zone picking parameters) from the cooperation with a German grocery retailer operating throughout Europe. We assume a planning horizon of $|T| = 13$ hours (e.g., 7 am until 8 pm) in which all deliveries have to be carried out. The distribution area is determined by a square of 200×200 kilometers, which represents the general setting at our retail partner, but also represents prevalent approaches in literature (see, e.g., Holzapfel et al. 2016; Schubert et al. 2018, 2021; Kuhn et al. 2021). The DC is located in the middle of the square, while customers are randomly assigned. We consider vehicles with a capacity of $Q = 60$ TUs. The order volume q_i per store ranges between 2 and 7 TUs. Further, the order volume is randomly split across available zones, where each order can be assigned to one zone or split across different zones. Only complete TUs are assigned. The number of pickers per zone depends on the workload, i.e., the required picking time for all assigned orders within each zone. More precisely, the total picking time per zone is divided by $0.3 \cdot \overline{T}$ and rounded up to the next integer to obtain the number of order pickers in the corresponding zone. Picking times per zone (p_{iz}) are calculated based on the volume of the corresponding suborders for each zone, with two minutes per TU. The setup time for preparing the picking process amounts to two minutes, the travel time to the intermediate storage to three minutes. Similarly, loading times (g_i) amount to two minutes per TU for each store order. The service times at stores (ψ_i) range between 15 and 30 minutes. Finally, the traveling cost is set at 1.2 EUR/kilometer, and the cost of vehicle usage ($\tilde{\omega}$) at 0.5 EUR/minute.

Problem classes. For our analysis, we generate different problem classes to examine various aspects of the problem, using a number of different settings for each characteristic. We generate 10 instances for each problem class, resulting in 6,480 instances in total. Table 6 summarizes the characteristics/settings of the test classes. We use two different time window scenarios. Short time windows refer to time windows of 90 minutes in length, while long time windows refer to 180 minutes. Please note that this defines the hard lower and soft upper bound of time windows. The hard upper bound for all stores equals the length of the planning horizon. For short time windows, the earliest time window starts after 240 minutes, whereas for long time windows it starts after 120 minutes of the planning horizon. This also determines the earliest starting times of vehicles. The

Table 6 Summary of problem characteristics

Characteristic	Specification	Comment
Number of orders	5, 10, 15, 20	Instance sizes for testing exact approach
	50, 100, 200	Instance sizes of practical relevance
Number of zones	1	–
	2	–
	3	–
Time window structure	1	Short time window scenario (90 min)
	2	Long time window scenario (180 min)
Number of vehicles	1	1.5 times the initial number of tours (small)
	2	2.0 times the initial number of tours (large)
Availability time of vehicles	0	In percent of the planning horizon
	25	In percent of the planning horizon
	50	In percent of the planning horizon
Intermediate storage size	25	In percent of the total volume
	50	In percent of the total volume
	75	In percent of the total volume
Target structure	1	Low tardiness costs ($0.5 \leq \overline{\omega}_i \leq 2.0$)
	2	High tardiness costs ($1.0 \leq \overline{\omega}_i \leq 4.0$)

shift in starting times further increases or decreases the pressure on the routing as less or more time is available for deliveries. Within both scenarios, time windows are assigned to stores according to a simple savings solution for vehicle routing. Moreover, the number of vehicles available is determined by 1.5 or 2.0 times the number of tours obtained within the initial savings solution for the routing. The availability time of vehicles indicates the time span within which vehicles become available at the DC. We consider three different availability settings: available from start (availability equals 0) or available within the first 25% or 50% of the planning horizon (using uniform distribution). The intermediate storage sizes vary between 25%, 50% or 75% of the total order volume. In line with this, the number of loading docks available is determined by dividing the given storage capacity by the capacity of vehicles (rounded to integer). Finally, we consider two cost scenarios for the pricing of tardy orders. In the first scenario, the cost parameter $\overline{\omega}_i$ varies between 0.5 and 2.0, while in the second it varies between 1.0 and 4.0 EUR/minute.

All algorithms have been implemented using C++ and the tests have been run on Intel Xenon E5-2697 processors. The model formulation for ZPRI_VRP (see Sect. 4) has been implemented in CPLEX Optimization Studio 12.9.

6.2 Comparison with exact approach

The ZPRI_VRP constitutes an NP-hard problem, and exact solutions are only possible for insignificant problem sizes when using standard software. We therefore

relaxed intermediate storage restrictions and loading operations to enable the comparison with an exact solution of the model (see Sect. 4) with CPLEX. This is done to analyze the solution quality of the iGVNS for small problem classes. We generated instances with up to 20 orders (see Table 6) and a vehicle availability of 0, 0.25 and 0.5, totaling 120 instances. We set the vehicle capacity at $Q = 30$ to encourage the creation of multiple tours. We further set a time limit of one hour per instance for CPLEX and five seconds for iGVNS. Please note that an increase in computing time for CPLEX was not possible due to working memory limitations. Table 7 summarizes the test results per problem class, i.e., ten instances were solved for each row. Column three shows the solution delta of CPLEX to iGVNS, i.e., $((f(\text{CPLEX})-f(\text{iGVNS}))/f(\text{iGVNS}))$. Columns four and five contain the optimality gap provided by CPLEX and the number of instances solved optimally by CPLEX. The last column shows the average computing time of CPLEX for each problem class.

Our results show that CPLEX struggles to find optimal solutions for instances with more than ten customers. Only two instances with 15 orders and none with 20 orders could be solved to proven optimality within a runtime limit of one hour. The average optimality gap by CPLEX for the 15 order instances amounts to 16.30%, and to 42.91% for 20 orders. Our iGVNS approach was able to match the CPLEX results, finding the optimal solution for all instances solved to optimality by CPLEX. In total, the iGVNS matches CPLEX results for 91 instances, while it generated superior results for the remaining 29 instances with an average improvement of 0.88% (15 order instances) and 3.58% (20 order instances). Our findings underline the complexity of our problem and the need for a tailored heuristic solution approach to solve practical relevant problem sizes.

Table 7 Comparison with the exact approach, 10 instances per test setting

Number of orders	Vehicle availability	Delta* iGNVS versus. CPLEX (%)	CPLEX gap (%)	Number of optimal solutions	Runtime CPLEX [s]
5	0	0.00	0.00	10	0.21
5	0.25	0.00	0.00	10	0.22
5	0.5	0.00	0.00	10	0.16
10	0	0.00	0.00	10	37.72
10	0.25	0.00	0.00	10	28.35
10	0.5	0.00	0.00	10	25.61
15	0	0.00	13.80	2	3600
15	0.25	1.00	17.51	0	3600
15	0.5	1.64	17.61	0	3600
20	0	1.70	43.21	0	3600
20	0.25	4.15	43.85	0	3600
20	0.5	4.88	41.67	0	3600

*Improvement of iGNVS compared to CPLEX

6.3 Application in retail practice

The ZPRI_VRP formulation is driven by an application in retail industry. We therefore present a case study based on the situation at our partner retailer. We leverage the data provided by our partner company and compare the results of the iGVNS to the prevailing approach in practice, i.e., routing first, picking second. This resembles the Seq_Prac approach (see Sect. 5.2).

Case description. We investigate the supply of markets with dairy products from a regional DC. The product categories considered are spread over three zones in the DC. The general problem structure is as described in Sect. 2. The structure of our case study fits into the following problem class defined in Sect. 6.1: {*time window structure = 1, number of vehicles = 2, availability time of vehicles = 50, intermediate storage size = 50, target structure = 1*}. We consider ten delivery days, i.e., ten different instances, with up to 125 orders.

iGVNS vs. Seq_Prac. The comparison shows that the iGVNS outperforms Seq_Prac by an average of 8.05% across all case instances, with a total cost improvement of up to 34.52%. This savings potential results from a significant reduction in tardy deliveries. While the routing and vehicle usage costs increase by an average of 16.85% and 9.01%, the tardiness costs are more than halved. An average of 37.6 markets are supplied with a delay using the iGVNS, while this figure is considerably higher in Seq_Prac solutions (59 markets). The average tardiness per market amounts to 15.7 minutes for iGVNS, and 27.5 minutes for Seq_Prac solutions. Moreover, 1.21% of markets were not supplied within the hard delivery time window in iGVNS solutions, in contrast to 6.21% of markets in Seq_Prac solutions. The number of late deliveries using Seq_Prac is therefore in line with the late deliveries reported at our case company. This tardiness can be reduced significantly using the iGVNS.

Impact of increasing intermediate storage. Subsequent to the overall performance of the iGVNS, the impact of the intermediate storage is a central aspect of our application. Limited intermediate storage affects picking and delivery processes on an operational level. Consequently, we analyze the impact of increasing storage size from about 50% of the daily delivery volume to 75% and 100%. Table 8 summarizes our results. All costs are normalized to the base case of 50% storage.

Table 8 iGVNS results for increasing storage sizes in the case study

Intermediate storage size	0.50	0.75	1.00
Total costs	100.00	91.04	83.51
Routing costs	100.00	98.07	95.19
Vehicle usage costs	100.00	99.00	97.23
Tardiness costs	100.00	56.04	24.35
Number of tardy orders	37.60	26.60	19.40
Number of unallowed tardy orders	1.50	0.30	0.00
Waiting time per picker [h]	1.43	0.51	0.00
Number of customers per vehicle	8.72	7.99	7.90

Increasing storage size has a positive effect on all cost variables. Total costs decrease by 8.96% and 16.49% when increasing the intermediate storage to 75% and 100%, respectively. These costs savings, however, originate mostly from an on-time supply of markets. This can be attributed to decreasing waiting times for order pickers when providing orders for delivery, i.e., the average provision time of an order decreases as well. This results in greater routing flexibility, making it easier to meet time windows. For an increase in size from 50% to 75%, the number of tardy orders decreases by 29.26%, and by 48.40% for an increase from 50% to 100% of the delivery volume.

Our case study shows that an integrated approach is needed to simultaneously address routing and picking operations, especially when the intermediate storage is restricted. The prevailing approach leads to higher total costs and lower service quality, with a high number of tardy deliveries. The resulting cost savings potential amounts to approximately 270,000 EUR (i.e., comparing objective values of Seq_Prac and iGVNS) per year and DC. Moreover, the size of the intermediate storage reveals a significant impact on overall costs and operations. It highlights the importance of considering the actual intermediate storage size in the planning. In the current retailer setup, the intermediate storage significantly impacts total costs and leads to limitations in the picking area. Relaxing the intermediate storage would provide a cost savings potential of approximately 600,000 EUR (i.e., comparing objective values of iGVNS solutions for a relaxed and restricted intermediate storage) per DC and year and would greatly increase order picking efficiency.

6.4 Extended experiments with the data generated

In our experiments with randomly generated data, we evaluate the performance of the solution approaches presented for different scenarios. We further assess whether an extended sequential approach (Seq_Iter) is able to improve the planning in retail industry. First, we assess the overall performance of the different approaches, followed by a detailed analysis with respect to problem characteristics. We would like to note that we generate our data sets randomly based on the given case study, i.e., assuming tight time restrictions. This means that feasibility is not ensured for single instances due to the given picking times, vehicle availability and delivery deadlines in combination with restricted intermediate storage.

6.4.1 Performance of iGVNS, Seq_Iter and Seq_Prac

Problem size. We examine the overall performance of the solution approaches introduced for different problem sizes. The analysis is based on the complete instance set (6480) and compares the final solutions. Please note that the comparison includes non-feasible solutions concerning inadmissible tardy orders to assess the performance across all problem classes considered on the same sample. We seek to find feasible solutions according to our model formulation (see Sect. 4) by steadily increasing the penalties for the violation of hard time windows during the search (see Sect. 5.1.5). Yet it may not be possible to reach a feasible solution at

the end of the search due to the tight time horizon. In order to nonetheless assess non-feasible solutions, we evaluate a final solution by ignoring the hard upper time window bound. This means that whenever a final solution violates the hard upper bounds (i.e., a non-feasible solution is obtained), we evaluate the solution by extending the soft time window penalty in a linear manner. This enables us to compare a sufficient number of instances. Otherwise no comparison would be possible for some problem classes. The computation times are limited to 20, 40 and 80 minutes for 50, 100 and 200 orders, respectively. Table 9 summarizes the comparison for each problem size.

The performance comparison shows that the iGVNS performs best across all instances. The iGVNS solution outperforms the Seq_Prac and Seq_Iter approach by 59.31% and 15.53% on average. Considering the two sequential approaches, Seq_Iter outperforms Seq_Prac by 39.69% on average. In general, the gap in solution quality rises with increasing problem sizes comparing the performance for Seq_Prac to the other approaches. However, for the instances with 200 orders, the gap decreases for some instances. This is particularly true for the comparison of Seq_Iter vs. iGVNS. While the gap is around 19% in the cases of 50 and 100 orders, the gap decreases to 7.62% in the case of 200 orders. This can be explained by the following two factors. First, the iGVNS requires additional computing time as the control logic (see Sect. 5.1.4) is called for more frequently for an increasing number of orders, especially when the intermediate storage is small. Second, the iGVNS achieves much more feasible solutions than both sequential approaches, and therefore its performance is underestimated using the given evaluation, i.e., penalizing non-feasible solutions only by extending the linear penalty. In total, the iGVNS reaches feasible solutions in around 77.10% of instances, while the sequential approaches only result in feasible solutions in 11.77% (Seq_Prac) and 22.61% (Seq_Iter) of instances. This emphasizes the need for integrated consideration of the given planning problem.

Value of integration. Additionally, we perform experiments to identify the impact of different parameters (see Table 6). This allows us to identify in which setting which solution approach is beneficial. The analysis is performed for instances with 100 orders (i.e., 2,160 instances), as this resembles the problem size of our case study. Table 10 summarizes the results across parameters. In contrast to the previous comparison, only feasible solutions are considered within the analysis.

Table 9 Comparison of solution approaches regarding total target value

Characteristic and specification		Deviation of target value [%]			Share of feasible instances [%]		
		Seq_Prac to iGVNS	Seq_Iter to iGVNS	Seq_Prac to Seq_Iter	iGVNS	Seq_Prac	Seq_Iter
Number of orders	50	52.09	19.09	24.75	99.63	27.04	45.83
	100	63.14	19.88	33.62	79.07	8.38	19.07
	200	62.68	7.62	60.71	52.59	0.23	2.92
	All	59.31	15.53	39.69	77.10	11.77	22.61

The results confirm the superiority of the integrated approach. The iGVNS outperforms the other approaches from 2.78% (Seq_Iter, target structure 1) to up to 31.06% (Seq_Iter, intermediate storage size 25%). Moreover, the iGVNS always reaches the best-known solution across all problem classes. However, performance and gap in solution quality vary greatly between the different classes. In less restricted scenarios (e.g., large intermediate storage, low penalty costs for tardy deliveries) sequential approaches perform well as they focus on the vehicle routing, which is then the main cost driver. Apart from the solution quality, we evaluate the performance of each approach with respect to the feasibility of the final solution. A non-feasible solution means that it was not possible to carry out some deliveries within the given planning horizon of 13 hours. In detail, the iGNVS is able to reach feasible solutions in around 80% of instances, while the Seq_Iter only succeeds in 19% of instances, and Seq_Prac in merely 8%. This again underlines the need for an integrated consideration of picking and routing operations when intermediate storage is restricted. Sequential approaches perform well for the routing aspect but fail to align the picking processes for the given time restrictions. Especially when the intermediate storage is more restricted or vehicles become available later, sequential approaches fail to provide feasible planning solutions. Finally, we would like to note that Seq_Prac has a smaller percentage deviation compared to the iGVNS than Seq_Iter in Table 10. This is due to the different sets of feasible solutions generated by the two sequential approaches. For instance, in cases where Seq_Prac fails to find a feasible solution, Seq_Iter may find a feasible solution at the expense of high routing costs. In a direct comparison regarding feasible solutions, Seq_Iter again leads to superior results.

Table 10 Impact of problem class parameters on total target value

Characteristic and specification		Percentage deviation of target value from			Share of feasible instances [%]		
		Seq_Prac to iGVNS	Seq_Iter to iGVNS	Seq_Prac to Seq_Iter	Seq_Prac	Seq_Iter	iGVNS
Number of zones	1	5.03	10.50	3.68	6.39	16.25	83.19
	2	5.90	9.84	1.70	7.08	16.94	78.89
	3	8.29	10.15	2.85	11.67	24.03	75.14
Time window structure	1	16.07	20.25	5.94	1.94	13.52	72.78
	2	5.57	4.91	2.51	14.81	24.63	85.37
Number of vehicles	1	5.20	9.01	1.90	9.07	18.06	73.52
	2	8.66	11.22	4.01	7.69	20.09	84.63
Vehicle availability	0	5.49	5.89	3.55	15.97	28.89	82.22
	25	6.29	13.35	0.14	7.22	17.36	79.86
	50	19.29	16.53	2.35	1.94	10.97	75.14
Intermediate storage size	25	–	31.06	–	0.00	2.78	40.00
	50	7.76	12.31	3.51	9.72	24.86	97.50
	75	6.18	7.57	2.27	15.42	29.58	99.72
Target structure	1	3.72	2.78	1.88	7.13	13.80	81.57
	2	9.06	14.31	3.14	9.63	24.35	76.57

In summary, an integrated approach is most beneficial if the alignment of routing and picking operations is essential for the distribution. This is particularly true for (i) a restrictive intermediate storage size, (ii) short time windows plus late vehicle start times, and (iii) a focus on reducing tardiness (i.e., in cases with higher penalties for late deliveries).

6.4.2 Detailed analysis of problem specifics

In our detailed analysis, we consider the impact of different parameter settings on the solution quality and solution structure. We therefore analyze the cost structure (share of distance, usage and tardiness costs), the average starting times of vehicles, and average blocking times for pickers. We use the 100 order instances feasibly solved by iGVNS for the detailed analysis (see above). An overview of the results obtained across problem classes is given in Table 11.

Number of picking zones. The use of multiple picking zones has a positive effect on the total cost value achievable. Compared to a single zone, the costs considered can be reduced by around 2% if two zones are used. However, the use of a third zone does not have an impact on the overall solution. The cost structure only changes slightly, reducing the share of tardiness costs, while usage costs increase. The reduction in overall costs and particularly tardiness costs can be attributed to decreasing starting times of vehicles. More zones enable the parallel picking of suborders and

Table 11 Detailed analysis of problem classes–iGVNS

Characteristic and specification		Total costs	Cost delta*	Cost share of			Average (in min.)	
							Vehicle start times	Blocking time for pickers
				Distances	Usage	Tardiness		
Number of zones	1	9.794		45%	41%	14%	220	64
	2	9.602	2%	45%	41%	14%	214	68
	3	9.605	2%	45%	41%	13%	213	68
Time window structure	1	9.724		44%	39%	17%	214	66
	2	9.609	1%	46%	43%	10%	218	67
Number of vehicles	1	10.260		42%	38%	19%	220	68
	2	9.074	13%	48%	44%	7%	212	64
Vehicle availability	0	9.228		46%	47%	7%	183	53
	25	9.349	– 1%	46%	42%	12%	209	64
	50	10.423	– 11%	43%	36%	21%	255	81
Intermediate storage size	25	12.637		38%	36%	26%	265	174
	50	8.354	51%	50%	45%	5%	197	22
	75	8.009	58%	51%	46%	3%	186	3
Target structure	1	8.902		48%	44%	8%	217	66
	2	10.431	– 15%	43%	39%	18%	215	67

*Delta versus first problem class of each characteristic

therefore the provision times are reduced. On the other hand, the zoning approach leads to increased blocking times of around 4 minutes per picker. We would like to note that the focus of our work is parallel picking for faster order availability and consideration of the additional complexity that occurs in real word applications. Different zoning strategies and possible savings in terms of workforce are not the focus of this research.

Time window structure. Larger time windows naturally decrease the pressure on timely deliveries. As a consequence, enlarging the time windows from 90 to 180 minutes leads to a cost reduction. The cost structure also changes significantly. In the case of larger time windows, the cost share for late deliveries decreases from 17% to 10%, while the distance (44% vs. 46%) and usage costs (39% vs. 43%) increase. As timely arrival becomes less critical, the average start time also increases by around 4 minutes.

Number of available vehicles. The number of available vehicles is essential for existing delivery options from the DC, even though not all vehicles are necessarily used. The iGVNS reduces distance and usage costs of vehicles, but there is no additional cost rate for the use of a vehicle as we assume a given fleet. If more vehicles are available, there are more options for tours and it is easier to reach customers in time. This results in a significant reduction in total costs of 13% if the vehicle ratio is increased from 1.5 to 2 times the tours of the routing approximation via the savings procedure. The reduction in total costs is driven by a strong decrease in tardiness costs (67%). The use of more vehicles further reduces the average starting times of tours (8 minutes) and blocking times of pickers (4 minutes).

Vehicle availability. Vehicle availability has a major impact on routing options as later availability postpones the possible start of tours. As a result, later availability causes significant cost increases. Compared to the availability from the start, costs increase by 11% if the vehicles just become available within the first half of the planning horizon (ratio 50%). Later availability limits the options for tours and tour starts and therefore increases distance, and—in particular—tardiness costs. This also results in a drastic increase in starting times (+72 minutes from ratio 0 to 50%) and blocking times for pickers.

Intermediate storage size. The intermediate storage size represents the core characteristic of the problem presented. Storage size dictates the rhythm of the distribution process, i.e., the interaction between picking and routing. The more restrictive the storage, the harder it is to align picking and routing operations and to reach customers in time. We show that intermediate storage of only 25% of total order volume has a significant impact on total costs. If 50% of the total order volume can be stored, total costs can be reduced by 51%, and by 58% if a storage size of 75% is available. The main driver for cost reduction is the decrease in tardiness. An increase in storage size goes hand in hand with a considerable decrease in vehicle starting and picker blocking times. While vehicles start after 265 minutes on average for the smallest storage size, start times are reduced to 186 minutes on average in the scenario with the largest storage size. Further, picker blocking times are reduced from 174 minutes to 22 minutes (50% storage) and to 3 minutes (75% storage). This means that there is almost no blocking time for pickers if the storage is sufficiently large.

Target structure. In the final problem classes, we analyze the impact of higher costs on tardy deliveries. As expected, higher penalties lead to an increase in total costs. In the classes tested, total costs rise by 15% on average if the range of penalties across customers doubles. The absolute tardiness costs increase by a factor of 2.5, and the share of tardiness costs increases from around 8% to 18%. The change in distance and usage costs is moderate, indicating that longer distances or higher usage are not able to compensate for the higher number of late deliveries. Again, the higher share of tardiness costs leads to earlier start times (− 2 min) and higher blocking times for pickers.

Summary. Alongside the need for an integrated approach (Sects. 6.3 and 6.4.1) for an application in practice, our extended experiments reveal the impact of main characteristics on total costs and solution structures. We summarize our findings regarding the distinct characteristics as follows:

- *Number of zones*: Parallel zoning with two zones leads to a reduction in total costs and earlier start times but increases idle times for pickers. Parallel zoning is therefore beneficial in the event that early start times are required due to time restrictions.
- *Time window structure*: Longer time windows reduce the pressure on timely deliveries, which results in lower total costs plus later starts of vehicles. Time window agreements have to be aligned with the system's capabilities.
- *Number of vehicles*: A larger delivery fleet enables a reduction in total costs as tardiness can be reduced and vehicles may start earlier for parallel deliveries. This also reduces the blocking times for pickers.
- *Vehicle availability*: The point in time at which vehicles become available is crucial for the fulfillment of deliveries in time. Late availability results in high costs as tour starts are postponed and picker blocking times increased, leading to high tardiness costs.
- *Intermediate storage size*: The storage size is the bottleneck in distribution and is the main driver for the alignment of picking and routing operations. Less storage leads to a significant increase in costs, vehicle start times and picker blocking times. The planning of sufficient storage capacities is thus crucial to enable efficient operations.
- *Target structure*: Higher penalties for tardy orders cannot be compensated by alternative routing options due to the existing limitations of the distribution system (time windows, storage capacity, vehicle availability). Total costs increase as a result.

In conclusion, total costs can be reduced significantly with a higher degree of freedom, i.e., when storage is greater, vehicle availability is earlier and penalties for tardy deliveries are lower. The results further show that the blocking times for pickers increase significantly for some problem classes (e.g., small intermediate storage size). Structural changes are possible in these cases. As we assume a constant number of pickers across problem classes with different specifications, the number of pickers may be reduced when the blocking times per picker increase.

7 Conclusion

The objective of our research is to highlight the need for integrated planning for routing and picking operations when intermediate storage is limited. This study introduces limited intermediate storage, vehicle availability times and zones picking operations to the research branch of integrated order picking and vehicle routing operations. We show the relevance of these problem characteristics by means of a case study with a major European food retailer. We present an extended model formulation and a tailored, integrated GVNS to address this complex planning problem. Our approach aims at minimizing total costs and we therefore consider driving, vehicle operation times and costs for tardy deliveries in our objective function. This enables us to achieve a realistic evaluation of the costs involved.

We use various experiments to analyze the integrated planning of picking and routing operations while respecting intermediate storage size. In particular, we compare the integrated approach proposed to two alternative solution approaches that are based on a sequential procedure. We show that the integrated planning approach outperforms sequential approaches by up to 60%, depending on the given problem characteristics. Moreover, we show that integrated planning is essential if the storage size is more restrictive. In scenarios with limited storage of around 25% of total demand, sequential approaches regularly fail to provide feasible solutions in terms of time restrictions and are therefore ineligible for effective planning in practice. The integrated solution approach proposed finds feasible solutions in 77.10% of instances, while a practically oriented and state-of-the-art sequential approach results in feasible solutions in only 11.77% and 22.61% of instances. All our experiments highlight the significant impact of restricted intermediate storage and the need for an integrated planning approach for efficient distribution in retail practice. In addition, we perform an extended analysis of the main characteristics of the problem and their impact on overall planning. Total costs can increase by up to 11%, depending on the availability of vehicles.

Future research. Our work considers a new problem variant and analyzes how this problem can be addressed, and which characteristics impact the decisions. However, there are various opportunities for future research in this context. First of all, we present an innovative, integrated solution approach. The solution approach applied is only a first step toward addressing this highly complex problem. Further alternative approaches need to be assessed and compared. Second, additional model extensions such as the consideration of a heterogeneous delivery fleet or the possibility of order batching could be valuable paths for future research as they offer further potential opportunities for cost savings. Our work focuses on distance, usage and tardiness costs. Other cost factors might be relevant depending on the application. We do not consider the impact of picking costs, for example. As we show, in some cases the blocking times for pickers are high and the number of pickers could potentially be reduced if picking costs are taken into account. In line with this, we consider multiple picking zones but assume that the number of pickers per

zone is predetermined and that picking times do not change as they purely depend on order sizes. In future research, the benefits of multiple picking zones should be assessed in more detail, especially with respect to time savings and potential reduction of the workforce.

Appendix A: iGVNS algorithmic framework

Data: Incumbent solution σ_{inc}, problem data, number of shaking neighborhood structures $\overline{\Phi}_{\text{GVNS}}$
Result: best solution σ_{best} and target value $f(\sigma_{\text{best}})$
while *termination criterion not met* **do**
  $\phi := 1$
  while $\phi \leq \overline{\Phi}_{\text{GVNS}}$ **do**
    determine σ_{temp} from the ϕth neighborhood of σ_{inc} ($\sigma_{\text{temp}} \in \Phi_{\phi}^{GVNS}(\sigma_{\text{inc}})$) randomly
    perform local search (VND(σ_{temp}))
    if $\sigma_{\text{temp}} \leq \sigma_{\text{best}}$ **then**
      update σ_{best} and σ_{inc}
      $\phi = 1$
    else if *acceptance criterion*($\sigma_{\text{temp}}, \sigma_{\text{inc}}$) *met* **then**
      update σ_{inc}
      $\phi = 1$
    else
      $\phi = \phi + 1$
    end
  end
end

Algorithm 1: General framework of iGVNS algorithm

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Archetti C, Feillet D, Speranza MG (2015) Complexity of routing problems with release dates. Eur J Oper Res 247:797–803

Archetti C, Jabali O, Speranza MG (2015) Multi-period vehicle routing problem with due dates. Comput Oper Res 61:122–134

Bansal V, Roy D (2021) Stochastic modeling of multiline orders in integrated storage-order picking system. Naval Res Logist 2021:1–27

Cattaruzza D, Absi N, Feillet D (2016) The multi-trip vehicle routing problem with time windows and release dates. Transport Sci 50:676–693
Chen Z-L (2010) Integrated production and outbound distribution scheduling: review and extension. Oper Res 58:130–148
Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 12:568–581
de Koster R, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order picking: a literature review. Eur J Oper Res 182:481–501
Dueck G, Scheuer T (1990) Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J Comput Phys 90:161–175
Fu Z, Eglese R, Li LYO (2008) A unified Tabu search algorithm for vehicle routing problems with soft time windows. J Oper Res Soc 59:663–673
Gallien J, Weber T (2010) To wave or not to wave? order release policies for warehouses with an automated sorter. Manuf Serv Oper Manag 12:642–662
Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130:449–467
Hansen P, Mladenović N, Brimber J, Pérez JAM (2010) Variable neighborhood search. In: Gendreau M, Potvin J-Y (eds) Handbook of metaheuristics, International series in operations research and management science, vol 146. Springer, New York, pp 61–86
Hemmelmayr VC, Doerner KF, Hartl RF (2009) A variable neighborhood search heuristic for periodic routing problems. Eur J Oper Res 195:791–802
Henke T, Speranza MG, Wäscher G (2015) The multi-compartment vehicle routing problem with flexible compartment sizes. Eur J Oper Res 246:730–743
Henn S (2015) Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. Flex Serv Manuf 27:86–114
Holzapfel A, Hübner A, Kuhn H, Sternbeck MG (2016) Delivery pattern and transportation planning in grocery retailing. Eur J Oper Res 252:54–68
Karimi IA, Reklaitis GV (1983) Optimal selection of intermediate storage tank capacity in a periodic batch/semicontinuous process. AIChE J 29:588–596
Karimi IA, Reklaitis GV (1985) Intermediate storage in non-continous processes involving stages of parallel units. AIChE J 31:44–52
Kovacs AA, Golden BL, Hartl RF, Parragh SN (2014) The generalized consistent vehicle routing problem. Transp Sci 49:796–816
Kuhn H, Schubert D, Holzapfel A (2021) Integrated order batching and vehicle routing: a general adaptive large neighborhood search algorithm. Eur J Oper Res 294:1003–1021
Moons S, Braekers K, Ramaekers K, Caris A, Arda Y (2019) The value of integrating order picking and vehicle routing decisions in a B2C e-commerce environment. Int J Prod Res 57:6405–6423
Moons S, Ramaekers K, Caris A, Arda Y, Moons S (2017) Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion. Comput Ind Eng 104:224–245
Moons S, Ramaekers K, Caris A, Arda Y (2018) Integration of order picking and vehicle routing in a B2C e-commerce context. Flex Serv Manuf J 30:813–843
Mouthuy S, Massen F, Deville Y, Van Hentenryck P (2015) A multi-stage very large-scale neighborhood search for the vehicle routing problem with soft time-windows. Transp Sci 49:223–238
Pirkwieser S, Raidl GR (2008) A variable neighborhood search for the periodic vehicle routing problem with time windows. In: Troyes, F. (ed) Proceedings of the 9th EU meeting on metaheuristics for logistics and vehicle routing
Potts CN, Strusevich VA (2009) Fifty years of scheduling: a survey of milestones. J Oper Res Soc 60:S41–S68
Ramaekers K, Caris A, Moons S, van Gils T (2018) Using an integrated order picking-vehicle routing problem to study the impact of delivery time windows in e-commerce. Eur Transp Res Rev 10:56
Rijal A, Bijvank M, de Koster R (2019) Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. Eur J Oper Res 278:752–771
Rijal A, Bijvank M, Goel A, de Koster R (2021) Workforce scheduling with order-picking assignments in distribution facilities. Transp Sci 55:725–746
Roodbergen KJ (2001) Layout and routing methods for warehouses. Ph.D. thesis, RSM Erasmus University, The Netherlands

Salhi S, Imran A, Wassan NA (2014) The multi-depot vehicle routing problem with heterogeneous vehicle fleet: formulation and a variable neighborhood search implementation. Comput Oper Res 52:315–325

Schmid V, Doerner KF, Laporte G (2013) Rich routing problems arising in supply chain management. Eur J Oper Res 224:435–448

Scholz A, Schubert D, Wäscher G (2017) Order picking with multiple pickers and due dates - simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. Eur J Oper Res 263:461–478

Schubert D, Kuhn H, Holzapfel A (2021) Same-day deliveries in omni-channel retail: integrated order picking and vehicle routing with vehicle-site dependencies. Naval Res Logist 68:721–744

Schubert D, Scholz A, Wäscher G (2018) Integrated order picking and vehicle routing with due dates. OR Spectr 40:1109–1139

Taillard ED, Badeau P, Gendreau M, Guertin F, Potvin J-Y (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. Transp Sci 31:170–186

Tappia E, Roy D, Melacini M, De Koster R (2019) Integrated storage-order picking systems: technology, performance models, and design insights. Eur J Oper Res 274:947–965

Toth P, Vigo D (2014) Vehicle routing: problems, methods, and applications, 2nd edn. Society for industrial and applied mathematics, MOS-SIAM series on optimization

van Belle J, Valckenaers P, Cattrysse D (2012) Cross-docking: state of the art. Omega 40:827–846

van Gils T, Ramaekers K, Braeker K, Depaire B, Caris A (2018) Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. Int J Prod Econ 197:243–261

van Gils T, Ramaekers K, Caris A, de Koster RBM (2018) Designing efficient order picking systems by combining planning problems: state-of-the-art classification and review. Eur J Oper Res 267:1–15

Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W (2012) A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Oper Res 60:611–624

Vidal T, Crainic TG, Gendreau M, Prins C (2014) A unified solution framework for multi-attribute vehicle routing problems. Eur J Oper Res 234:658–673

Witt A, Voß S (2007) Simple heuristics for scheduling with limited intermediate storage. Comput Oper Res 34:2293–2309