Submitted to *Management Science*
manuscript MS-OPT-18-01682.R4

Optimal picking policies in e-commerce warehouses

Maximilian Schiffer
TUM School of Management & Munich Data Science Institute, Technical University of Munich, 80333 Munich, Germany, schiffer@tum.de

Nils Boysen
Chair of Operations Management, Friedrich Schiller Universität Jena, 07743 Jena, Germany, nils.boysen@uni-jena.de

Patrick S. Klein
TUM School of Management, Technical University of Munich, 80333 Munich, Germany, patrick.s.klein@tum.de

Gilbert Laporte
HEC Montréal, Montréal H3T 2A7, Canada & School of Management, University of Bath, Bath BA2 7AY, United Kingdom, gilbert.laporte@cirrelt.ca

Marco Pavone
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94035 USA, pavone@stanford.edu

In e-commerce warehouses, online retailers increase their efficiency by using a mixed-shelves (or scattered storage) concept, where unit loads are purposefully broken down into single items, which are individually stored in multiple locations. Irrespective of the stock keeping units a customer jointly orders, this storage strategy increases the likelihood that somewhere in the warehouse the items of the requested stock keeping units will be in close vicinity, which may significantly reduce an order picker's unproductive walking time. This paper optimizes picker routing through such mixed-shelves warehouses. Specifically, we introduce a generic exact algorithmic framework that covers a multitude of picking policies, independently of the underlying picking zone layout, and is suitable for real-time applications. This framework embeds a bidirectional layered graph algorithm which provides the best known performance for the simple picking problem with a single depot and no further attributes. We compare three different real-world e-commerce warehouse settings that differ slightly in their application of scattered storage and in their picking policies. Based on these, we derive additional layouts and settings that yield further managerial insights. Our results reveal that the right combination of drop-off points, dynamic batching, the utilization of picking carts, and the picking zone layout can greatly improve the picking performance. In particular, some combinations of policies yield efficiency increases of more than 30% compared with standard policies currently used in practice.

Key words: mixed-shelves warehouse, order picking policies, picking zone layout, dynamic programming
History:

1. Introduction

A paradigm change in the retail business, caused by rapidly growing e-commerce (Statista 2017, Cui et al. 2019), makes today's warehouses the focus of efficient operations in business-to-customer (B2C) markets. While warehouses have traditionally been viewed as remote, negligible, and cost-efficient planning components, today's warehouses have evolved into technology-enriched logistics facilities that play a key role in retail supply chains. For companies of any significant size, the warehouse of today and of the future performs several of the functions that conventional stores have previously managed. Consequently, firms' attention to innovative concepts and efficient warehouse operations is steadily increasing, as warehouses are seen as a key success factor in the highly competitive e-commerce market. In these warehouses, efficient daily operations are central to running a profitable e-commerce business and to fulfilling customer expectations such as same-day deliveries, such that efficiently routing pickers constitutes a crucial determinant of profitability.

In e-commerce warehouses, two fundamentally different concepts dominate: robotized parts-to-picker warehouses, where (KIVA) robots bring man-high shelves to stationary pickers (Azadeh et al. 2018), and human-operated picker-to-parts warehouses, in which pickers traditionally walk through the shelves and collect stock keeping units (SKUs). These two concepts entail a trade-off between investment costs, scalability, and efficiency. *Parts-to-picker warehouses* have high investment costs that increase significantly with the number of robots but are highly efficient when operated at their maximum throughput. They are hardly scalable to rare events during which the workload increases significantly, e.g., on Black Friday or for Cyber Monday sales. *Picker-to-parts warehouses* have low investment costs and are less efficient than robotized systems. In contrast, these warehouses allow—by hiring additional temporary staff—the necessary flexibility in peak times. Hence, major players in the industry agree that picker-to-parts warehouses should remain in parallel to robotized warehouses to provide flexibility.

With standardized third-party robotized warehouse solutions across companies, increasing the efficiency of picker-to-parts warehouses becomes even more crucial for online retailers who vie to derive a competitive advantage. In traditional picker-to-parts warehouses (with unit loads kept together), the storage assignment plans result in excessive unproductive walking for the pickers, corresponding to as much as 50% of their working hours (De Koster et al. 2007). Many online retailers in the B2C segment apply the mixed-shelves or scattered storage concept (Weidinger and Boysen 2018) to increase the efficiency of such warehouses. Under this storage assignment policy, incoming loads of SKUs are purposefully broken down into single items scattered over all parts of the warehouse, thus increasing the likelihood that, given the volatile demands of the final customers, items finally ordered together will be in close vicinity somewhere in the warehouse (Boysen et al. 2019a). However, competitive routing algorithms that determine picker paths to collect SKUs remain crucial to an efficient exploitation of the scattered storage in e-commerce warehouses, and to the reduction of unproductive picker walking time. In particular, large e-commerce warehouses subdivide their order fulfillment process into three stages, i.e., order picking, intermediate storing of picked bins, and order accumulation and packing (see Section 2.1). While the latter two stages are rather standardized and can largely be automated, order picking remains the most crucial and labor-intensive stage and is, therefore, the focus of this paper.

Today's e-commerce warehouses consist of picking zones, i.e., limited blocks of shelves in which pickers operate (see Figure 1). These zones have a rectilinear layout, which means that the pickers always move according to a Manhattan metric, with shelves placed along parallel aisles and one or several drop-off points at which pickers can hand over items. Online retailers apply batch picking, i.e., they unify multiple picking orders into a single picklist (Boysen et al. 2019a) which is then processed by a picker in a dedicated zone. The picker starts with a manual picking cart and a picklist at a drop-off point and returns to it once the picklist is completed. Despite the regular rectilinear layout and the rather simple picking process, picking zones may differ in the number of cross-aisles and aisles they contain. Although most companies rely on the scattered storage principle and agree on efficient picking as a key to successful operations, there is no consensus on which general attributes a good picking policy and a picking zone layout should possess. This paper focuses on four main attributes which have been identified as most frequently implemented in practice in the recent survey of Boysen et al. (2019a):

i) *a variable multi-block layout* which differs in terms of the number of aisles and cross-aisles;
ii) *multiple drop-off points* which increase the number of points in a picking zone where completed bins can be handed over to a central conveyor system;
iii) *dynamic batching policies* which allow to drop a finished picklist early while processing several picklists in parallel by equipping the picking cart with multiple bins; and
iv) *cartless subtours* which allow pickers to intermediately park their clumsy cart in order to pick a few items on a subtour much faster.

Several site visits that we have made in European e-commerce warehouses have confirmed that these policies and layout options are applied in various combinations (see Section 2.2), and benefits can be achieved through optimization. However, performance gains come at the price of additional investments, which may differ for each setting. Hence, we evaluate the performance impact of the picking policy and layout options, which allows us to quantify each measure's improvement potential without relating it to varying and hence hardly assessable investment costs. To set our

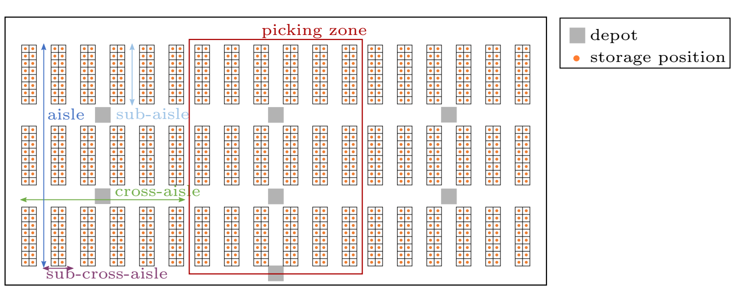


Figure 1 Example of the storage area of an e-commerce warehouse and a picking zone.

study apart from recent research, we first briefly review the related literature in Section 1.1, before further detailing the contributions of our work in Section 1.2.

1.1. Literature review

Efficient order processing in warehouses and distribution centers has been vividly discussed in recent years. For an overview of the rich body of literature on traditional picker-to-parts warehouses, modern robotized warehouses, and the peculiarities of warehousing in the e-commerce era, we refer to the surveys of De Koster et al. (2007), Azadeh et al. (2018), and Boysen et al. (2019a). In the following, we only survey the picker routing literature related to the planning tasks outlined above.

Picker routing starting and ending at a central drop-off point corresponds to the traveling salesman problem (TSP). In their seminal paper, Ratliff and Rosenthal (1983) showed that in single-block warehouses consisting of parallel aisles with cross-aisles at the front and back, the particular structure of the distance matrix allows solving the resulting TSP in polynomial time by dynamic programming. Roodbergen and De Koster (2001) extended this approach to two-block warehouses with an additional middle aisle, and De Koster and Van der Poort (1998) considered a handover of complete orders at the start and end of each aisle. Recently, Pansart et al. (2018) generalized the approach of Ratliff and Rosenthal (1983) to multi-block warehouses using the polynomial-time algorithm of Cambazard and Catusse (2018) for rectilinear Steiner tree problems in the plane. As can be seen, exact algorithms for the picker routing problem are rare and limited to special cases.

Several heuristics have been proposed for the picker routing problem itself and for some of its variants, e.g., picker routing combined with selecting pick positions or with zoning and batching policies. For a rich overview of these heuristics, we refer the reader to De Koster et al. (2007). While fundamental work including performance analysis was carried out by Hall (1993) and Petersen (1997), De Koster and Van der Poort (1998) provided comparisons between exact and heuristic algorithms for specific layouts. The more recent approximate algorithms adapt competitive TSP heuristics to the picker routing problem (Theys et al. 2010). Gue et al. (2006), Hong et al. (2012a), and Chen et al. (2013) have all focused on blocking aspects in narrow aisles. Given the dedicated zoning policy and layout of modern e-commerce warehouses, this blocking issue is negligible nowadays. Only one publication (Goetschalckx and Ratliff 1988) exists on pickers with different travel modes, i.e., pickers who are allowed to park their cart and perform cartless subtours on foot. However, that paper assumes that a picker moves the vehicle along a single aisle, but starting and stopping the vehicle to retrieve items on foot consumes additional time. As can be seen, no heuristic exists that captures the generic nature of our problem.

Weidinger et al. (2019) showed that the integrated selection of pick positions for a specific demand in a mixed-shelves warehouse makes the picker routing problem strongly NP-hard, even in elementary block-shaped warehouses with parallel aisles. Algorithms for this problem class were proposed by Daniels et al. (1998), Weidinger (2018), and Weidinger et al. (2019). These authors decompose the problem by fixing the pick positions' selection and treating the second-stage routing problem heuristically. Overviews of zoning and batching policies can be found in De Koster et al. (2007) and Boysen et al. (2019a). Recent contributions to this field by Bozer and Kile (2008), Henn and Wäscher (2012), Hong et al. (2012b), and Zulj et al. (2018), have focused on heuristics for a static batching policy. In practice, the two planning tasks of selecting pick positions and batching and zoning are decoupled from the picker routing decision. Recently, Weidinger (2018) gave proof to this common practice and showed that solutions derived in this fashion yield optimality gaps well below one percent compared to integrated planning. Hence, we exclude these problems from our real-world study, and assume respective decisions to be made at an upstream decision stage.

Concluding, no exact or heuristic algorithm exists that captures the generic nature of our problem, namely i) *a variable multi-block layout*, ii) *multiple drop-off points*, iii) *dynamic batching policies*, and iv) *cartless subtours*. In particular, requirements ii)–iv) have not yet been incorporated in any exact or heuristic algorithm to the best of our knowledge.

1.2. Contributions

In this paper, we close the gap in the literature outlined in Section 1.1 by developing the first generic exact algorithm that can handle the four features just mentioned. Its computational requirements must be sufficiently low to allow for real-time application in practical settings. Given this, the methodological and managerial contribution of this paper is threefold. First, we develop an exact algorithm for picker routing capable of capturing the four main attributes identified by Boysen et al. (2019a). Second, we provide additional preprocessing techniques that allow the real-time use of this algorithm in real-world applications. Third, based on this methodological framework, we conduct different experiments to benchmark several picking policies and layout options on realistic data sets extracted from three corporate case studies.

Our algorithm provides a new state of the art for picker routing problems. In particular, we outperform the previously best known exact algorithm (Pansart et al. 2018) for single picklists in multi-block warehouses in terms of computational time, and we obtain better scalability for large-size instances with up to 11 cross-aisles. Besides, our algorithm is significantly more general as it can solve picker routing problems with multiple picklists, dynamic batching strategies, multiple drop-off points, and cartless subtours. We use this algorithm to identify the benefits of all considered picking policy design options utilizing a full-factorial experiment. Specifically, we show that the level of picklist dispersion, i.e., a picklist's warehousing area which results from the selection of more or less dispersed picking positions during upper level planning tasks, strongly influences strategic design and operational decisions.

1. At the strategic level, we show that increasing the number of cross-aisles can yield efficiency increases of more than 20% in the case of dispersed picklists. However, for clustered picklists, the reduction potential is much smaller, and increasing the number of cross-aisles may even increase the total picker walking time. The deployment of multiple drop-off points shows decreasing marginal utilities for large numbers of drop-off points.
2. At the operational level, we show that dynamic batching yields significant efficiency improvements of up to 9% for clustered picklists but has only negligible effects in the case of dispersed picklists. In contrast, cartless subtours yield efficiency improvements of up to 11% for dispersed picklists but have a negligible effect in the case of clustered picklists.
3. Combining the right operational and strategic design decisions can lead to efficiency improvements of up to 31% for clustered picklists and of up to 11% for dispersed picklists.

1.3. Organization of the paper

The remainder of this paper is structured as follows. Section 2 further details the fulfillment process in an e-commerce warehouse, details the settings of picking policies we have observed in practice, and provides a formal definition of the planning problems at hand. Section 3 develops our algorithm. We present extensive computational results in Section 4. Section 5 concludes this paper and provides several managerial insights. All proposition proofs are provided in Appendix A.

2. Problem description

This section introduces our planning problem. We first detail the order fulfillment process in e-commerce warehouses to show the integration of our planning problem into daily operations. We then discuss the settings of picking policies that we have observed in practice, before formally introducing the generalized order picking problem (GOPP).

2.1. Order fulfillment process

For conciseness, we exclude the ingoing flow of the warehouse and focus on the outgoing flow. Once items are stored and scattered around the warehouse, the order fulfillment process in a mixed-shelves warehouse consists of three basic steps.

Order picking: First, customer orders must be picked from their respective storage positions. Human pickers, each equipped with a small picking cart that carries small bins, collect items into these bins. A picker receives empty bins and picklists, each associated with a bin, at a drop-off point, strides (directed by a hand-held scanner or a pick-by-voice system) through the warehouse, collects the requested items from the shelves, and puts them into their respective bin on the cart. Once the picker has completed one or multiple bins, she hands the bins over to the central conveyor system at a drop-off point and starts processing the next picklists. This process is typically executed under a zoning and batching policy (De Koster et al. 2007). Instead of storing all items in a single monolithic area, online retailers subdivide their vast shop floors into multiple zones to allow for parallel processing of customer orders. A picker operates exclusively in one of these zones and only picks the part of an order stored in her assigned zone. By parallelizing the picking process in this fashion, order pickers traverse smaller areas of the warehouses to reduce their unproductive walking time. A further reduction of unproductive walking time can be achieved by unifying multiple orders into batches that are jointly retrieved on the picker's tour through her zone. This results in bins filled with partial orders for multiple customers.

In e-commerce, customer orders have different priorities and varying deadlines as some customers may participate in priority delivery programs. Directly considering these priorities leads to unrealistically long planning horizons of several hours that may even exceed the time needed to handle the orders known beforehand. In practice, warehouse operators consider the priority of orders in line with the order batching and picklist sequencing at an upstream planning stage before giving a picklist sequence, which comprises the most urgent orders for a short planning horizon, to a picker. This sorted list sequence comprises the picklists for the most urgent orders. The order waves that are not yet included in this planning horizon are more likely to be timely processed if the current set of picklists is assembled as fast as possible.

Intermediate storing of picked bins: Once handed over to the central conveying system at a drop-off point, these bins are intermediately stored until all those belonging to the same wave of customer orders have arrived from all zones. Different storage devices exist. Some operators apply lane-based systems where bins of a wave are channeled in a conveyor queue. Others apply automated storage and retrieval systems, where bins are stored in crane-operated aisles of high-bay storage racks (Boysen et al. 2018). In smaller warehouses, loop-based systems where bins circle until a wave is complete may be applied (Gallien and Weber 2010).

Order accumulation and packing: Finally, a wave of bins released from intermediate storage arrives in the order accumulation area. Here, the incoming items are sorted according to customer orders. Some operators apply so-called put walls for this task, where human logistics workers place items on small shelves, each temporarily dedicated to a specific customer order (Boysen et al. 2019b). On the other side of the wall, packers receive completed orders, pack them into their shipping cartons, and forward them via conveyors to their dedicated trailers. Other warehouses apply automated belt sorters, e.g., tilt-tray or cross-belt sorters, where items are collected in packing stations arranged along the belt (Boysen et al. 2018).

Both *intermediate storing* and *packing* are process steps that can easily be standardized and organized efficiently. However, *order picking* includes unpredictable components because it depends on incoming customer orders and requires the largest fraction of the workforce. Therefore, an efficient routing algorithm plays a crucial role in ensuring efficient order fulfillment.

The inputs needed for creating route plans are the picklists, a picklist sequence, and the assignment of SKUs to picklists. Integrated problems that combine order batching or sequencing with picker routing have been discussed (Weidinger et al. 2019), but while they provide challenging research questions, their solutions are less relevant in practice. Indeed, operators typically solve the batching and sequencing problems at an upper level, decoupled from the picker routing, for the following reason: both problems heavily depend on the incoming orders' priorities. Those orders with the closest due date of their associated outbound trucks are selected as the wave of orders to be processed next. Therefore, all orders of the current wave are urgent, and there is no need to further differentiate between their priorities during order picking. Recent research has corroborated this common practice. Weidinger (2018) has shown that in practice selecting the shelves from which each item is to be picked in a scattered storage warehouse can be solved independently of the picker routing problem. Indeed, selecting shelves such that the current picklist's warehousing area is minimal and handing these pick positions over to picker routing leads to near-optimal solutions with optimality gaps well below one percent (Weidinger 2018).

2.2. Picking policies

In this paper, we develop a generalized exact algorithm for picker routing in e-commerce warehouses. We create different settings based on different order-picking policies that we have observed in practice in mixed-shelves warehouses.

Amazon Europe: In Amazon's distribution centers in Bad Hersfeld (Germany) and Poznan (Poland), we have observed the following setting. Amazon uses multiple drop-off points to a central conveyor system to hand over completed bins. Pickers often pass by these local drop-off points during a tour, which offers the additional flexibility of dynamic batching. Instead of picking batch after batch with intermediate returns to a central drop-off point in a static manner, pickers can hand over just a subset of completed bins at a local drop-off point and retrieve new empty bins for subsequent orders. Thus, Amazon's picker routing procedure additionally considers the dynamic batching of an order set by inserting visits at different drop-off points in the tour where the picker hands over subsets of completed bins and starts handling new orders.

Hermes Group: This is Germany's second-largest postal service provider. Its distribution center in Haldensleben (Germany) operates full-service order fulfillment for one of Europe's largest online and catalog fashion retailers. Hermes's order picking process is as follows. Each picker is equipped with a picking cart with up to four bins, each having a capacity of about 20 to 30 items. Pickers operate in fixed zones, each having a single drop-off point where new bins and the next picklist are retrieved and, finally, completed bins are handed over to the central conveyor system. Hence, each picker tour starts and ends at a central drop-off point, and for a given batch of (four) picking orders, one seeks a tour through the respective zone of the warehouse, such that all items on the picklist can be retrieved, and the length of the tour is minimal.

Zalando: The online fashion retailer Zalando uses a different system at its distribution center in Erfurt (Germany). It applies static batching with a single drop-off point in each picking zone. Since the bins are relatively large and heavy when filled, picking carts are clumsy and inflexible. Hence, the pickers are much faster without the cart. Thus, a picker may park her cart, pick up to six items from close-by shelves, and carry them back to the cart. An optimization procedure for the Zalando case must determine whether a walk between two successive visits of a picker tour should be executed with or without a cart. This decision has to consider the limited capacity of items a picker can carry on her arms and the varying walking speeds with and without a cart.

Based on these observations, we classify picking policies according to three characteristics: i) *static vs. dynamic batching*, ii) *single vs. multiple drop-off points*, and iii) *single vs. multiple travel modes (i.e., cartless subtours)*. Certain combinations of these characteristics reflect the real-world picking policies detailed above. In our computational studies, we aim at a full factorial design to evaluate each degree of freedom's potential in designing picking policies. Hence, we add missing policies even if we have not yet observed them in practice. Table 1 shows the different picking policies with increasing degrees of freedom from left to right. As can be seen, the Amazon (A) policy already covers most degrees of freedom, while the Hermes (H) and the Zalando (Z) policy remain at the bottom line with respect to degrees of freedom.

2.3. A generalized order picking problem

Given the multitude of picking policies described in Section 2.2, we now formally introduce the GOPP.

Solution representation and notation: To represent a solution Π, we model positions in the picking zone (Figure 1) as vertices, such that a vertex is either a drop-off point ($c \in \mathcal{D}$), a crossing between an aisle and a cross-aisle ($c \in \mathcal{C}$), or a storage position ($c \in \mathcal{P}$) that contains varying amounts of different SKUs. We note that a solution will include only a subset of storage position vertices $\mathcal{P}' \subseteq \mathcal{P}$, that correspond to positions where items have to be picked. Analogously, only

Table 1 Picking policies resulting from different attributes.

Policy	i (H)	ii (Z)	iii	iv	v	vi	vii (A)	viii
static (SB) / dynamic (DB) batching	SB	SB	SB	SB	DB	DB	DB	DB
single (SD) / multiple (MD) drop-off points	SD	SD	MD	MD	SD	SD	MD	MD
no cartless subtours (NS) / cartless subtours (CS)	NS	CS	NS	CS	NS	CS	NS	CS

drop-off point vertices and cross-aisle vertices necessary to complete a tour are included in the respective solution. At the operational level, an ordered set $\mathcal{L}$ of picklists l denotes which position $c \in \mathcal{P}$ must be visited to pick up SKUs. Recall that the size and pick-up position of a given SKU is decided at a preceding planning level to efficiently split customer orders into picklists and avoid double picking by competing pickers. Hence we represent a single picklist $l = \{c_1, ..., c_n\}$ as a finite set of n storage positions that must be visited. Across multiple picklists, n can vary as, for example, prioritized lists might be designed with fewer items on an upper planning level.

Besides, we use the set $\mathcal{L}_i^{\mathrm{a}}$ to track active picklists at each position i of a route. Depending on the picking policy, a picker processes a single picklist or multiple picklists in parallel such that $|\mathcal{L}_i^{\mathrm{a}}|$ is limited. Each picklist must be collected in a separate bin. Thus, we can measure the *picking cart capacity* κ, i.e., the number of possible parallel-processed picklists, by the number of bins. In practice, standardized bins are a prerequisite for fail-safe transport on picking carts and conveyors. Note that our assumption remains valid even for cases with differently sized bins since the upstream planning stage avoids conflicting bins while defining the sequence of picklists.

A picker is accompanied by a cart equipped with standardized bins to collect orders. For certain picking policies, the picker may temporarily leave her cart to pick a certain number of SKUs during a *cartless subtour.*

Definition 1 (cartless subtour). A cartless subtour of the picker starts at the i^{th} stop of the tour where the picker leaves her cart and ends at stop k where she picks up her cart again at the same position such that $c_i = c_k$. In addition, a cartless subtour is limited to a certain number of items K, i.e., the maximum number of items a picker can carry without cart support. A picker may start a subtour at the end of or at any position in a sub-aisle. A subtour may not span across more than one sub-aisle.

A solution Π is an ordered set that defines the successive visits of the picker in $\mathcal{V} = \mathcal{D} \cup \mathcal{C} \cup \mathcal{P}$. Each element of Π is a pair $\pi_i = (c_i, \mathcal{L}_i^{\mathrm{a}})$. The pair $\pi_0 = (c_0, \mathcal{L}_0^{\mathrm{a}})$ at the very first sequence position refers to the initial position of the picker, such that c_0 is fixed to the picker's starting point. This ordered set defines a route through the picking zone such that a sequence of picklists $\mathcal{L}$ is completed.

Objective function: The main factor influencing the total picking time at this operational stage is the picker walking time as most other parts of the picker's time, e.g., the time to retrieve items from a shelf is fixed once a picklist is given. Hence, the GOPP's objective minimizes the total picker walking time, considering the picker's walking time τ_{c_{i-1},c_i} between consecutive vertices in Π

$$Z(\Pi) = \sum_{i \in \{1,...,|\Pi|-1\}} \tau_{c_{i-1},c_i}. \tag{1}$$

Constraints: To properly define all constraints for the GOPP, which differ depending on the type of batching, on the available travel modes, and on the number of drop-off points, we introduce the notion of *active intervals.*

Definition 2 (active interval). Given a solution Π, a picker starts a not yet processed picklist l at a drop-off point preceding the first storage position visit related to l. Analogously, she hands over a completed picklist l at a drop-off point succeeding the last storage position visit related to l. During the time between these two drop-off point visits, we refer to a picklist as active and to the complete time span during which a picklist is active as its active interval.

Then, the constraints for the GOPP are as follows:

i) The cart capacity κ is limited and $|\mathcal{L}_i^{\mathrm{a}}| \leq \kappa$ must hold for any pair π_i.

ii) Once a list starts being processed by a picker, it must be finished before a new picklist can be processed in the respective bin, i.e., each picklist has exactly one coherent active interval.

iii) A picklist l cannot be completed before all visits of l are covered in its active interval.

iv) $\mathcal{L}$ is processed in its given order.

iv.s) For static batching, the active intervals of picklists either completely overlap or are completely disjoint.

iv.d) For dynamic batching, (iv.s) is relaxed. This allows handing over merely a subset of already completed active orders when stopping at a drop-off point.

v) Bins can only be handed over at a drop-off point if the cart carrying them is also at the drop-off point, i.e., no drop-offs during cartless subtours are allowed.

vi) All picklists must be completed, which is fulfilled when each picklist has exactly one active interval.

vii) If cartless subtours are allowed, only feasible subtours occur, such that the picker's capacity for carrying items is not exceeded, leaving and returning of hand-carried items refer to the same parking position of the cart, and the subtour occurs within one sub-aisle.

Among all feasible solutions fulfilling these constraints, the GOPP seeks a solution Π^* that minimizes the objective function (1).

3. Methodology

In Appendix B we prove that the GOPP is strongly NP-hard. However, a special case of the GOPP can be solved in polynomial time. Specifically, for a setting with a single picklist, a single travel mode, and a single drop-off point, the rectilinear layout characterizing the GOPP (Figure 1) limits the number of possible ways to traverse a sub-aisle (see Section 3.1). Accordingly, after summarizing the findings of previous work (Section 3.1), we first develop a skeleton for our algorithmic framework that efficiently solves the GOPP, limited to a single picklist, travel mode, and drop-off point, by using a *layered graph algorithm* (LGA) (Section 3.2). We present a complementary integer linear program (ILP) based skeleton in Appendix C. We then derive a unified framework where we incorporate this specific version of the GOPP as a subproblem such that we can incorporate the skeletons to solve all variants of the GOPP based on branching and pruning rules (Section 3.3).

3.1. State of the art

A reduced version of the GOPP constitutes a special case of the rectilinear TSP which allows us to exploit some properties of this problem. Hence, for the sake of completeness, we summarize state-of-the-art developments in two fields: exact algorithms for picking problems and recent enhancements of exact algorithms for the rectilinear TSP.

Exact algorithms for order picking are still rare (cf. Section 1.1). To the best of our knowledge, only three algorithms exist, all based on the work of Ratliff and Rosenthal (1983) who presented a first polynomial algorithm for a specified warehouse layout, namely a warehouse without interspersed cross-aisles. In a nutshell, the NP-hardness of the general TSP is removed since it can be proven that at most two arcs between adjacent vertices exist in a solution. This algorithm scales linearly with the number of sub-aisles since its complexity is significantly reduced by Proposition 1, which introduces the concept of *transitions* to denote a picker's movement through a sub-aisle (a, b).

Example 1 (Using transitions to model picker movements in sub-aisles). Given a sub-aisle (c, c') that starts at c and ends at c', a transition denotes a possible way for a picker to travel through (c, c') in order to collect all items that she must pick in this sub-aisle. Figure 2 shows examples of such transitions for a sub-aisle (c, c').

Proposition 1. [adapted from Ratliff and Rosenthal (1983)] *Let $G = (\mathcal{V}, \mathcal{A})$ be a graph representation of a rectilinear picking zone. Then, an optimal picker tour, i.e., the shortest tour that covers every position with an item that must be picked, can be represented by using only six different transitions for a sub-aisle (c, c') and three different transitions for a sub-cross-aisle.*

Figure 3 illustrates the transitions of Proposition 1 for a general representation of (c, c') with two (optional) stops in the sub-aisle. Besides the transitions already detailed in Figure 2, these transitions contain an empty transition (I), as some sub-aisles that do not contain items that must

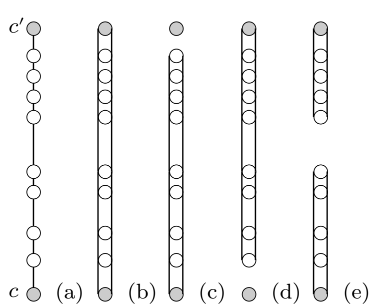


Figure 2 **Examples of transitions for a sub-aisle.**

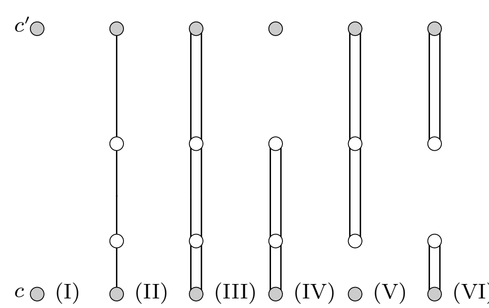


Figure 3 **Sub-aisle transitions for optimal paths in G.**

be picked may not be visited in an optimal solution. Summarizing, the complete set of transitions contains a non-traversed sub-aisle (I) and a complete traversal in one (II) or two (III) directions. Furthermore, a sub-aisle may be partially traversed with slopes from one (IV,V) or both (VI) directions. For transition (VI), the split between both slopes is always given by the longest distance between two stops in the same sub-aisle if more than two stops are present in the sub-aisle. Although this split criterion may not be unique, it does not affect the solution's global optimality since all split options have the same cost. For cross-aisles in which the picker does not pick up items, only transitions I–III, from here on referred to as standard transitions, are necessary.

Based on this rationale, Roodbergen and De Koster (2001) and Pansart et al. (2018) have proposed exact algorithms for warehouse layouts with three or more cross-aisles. However, these algorithms are limited in their applicability to the GOPP since multiple picklists, parallel processing of picklists, multiple drop-off points, and multiple travel modes are still not considered.

Cambazard and Catusse (2018) proposed an exact algorithm based on finding an optimal tour for a rectilinear TSP with a set of customers $\mathcal{V}$, by solving a Steiner variant of this problem on the *Hanan graph* $H(\mathcal{V})$ with terminal vertices $\mathcal{V}$. Based on Proposition 2, they decompose the problem and develop a layered graph algorithm that uses a dynamic program to explore $H(\mathcal{V})$ iteratively to find an optimal tour.

Definition 3 (Hanan graph). Let $\mathcal{U}$ be a finite set of vertices in a plane. We first construct a grid, made up of vertical and horizontal lines through each point of $\mathcal{U}$, commonly known as a Hanan grid (Hanan 1966). The resulting intersections define a second set of vertices $\mathcal{S}$ which includes $\mathcal{U}$. The Hanan graph $H(\mathcal{U}) = (\mathcal{S}, \mathcal{E})$ formally defines this grid and consists of the vertex set $\mathcal{S}$ and an edge set $\mathcal{E}$, which contains all edges linking two consecutive vertices of $\mathcal{S}$ in the horizontal or vertical direction.

Proposition 2. [adapted from Cambazard and Catusse (2018)] *Let T^* be an optimal tour in $H(\mathcal{V})$. Then, there exists a planar separator S of $H(\mathcal{V})$ that decomposes T^* into two partial tours T_l^* and T_r^*.*

Definition 4 (planar separator). For a Hanan graph $H(\mathcal{U}) = (\mathcal{S}, \mathcal{E})$, we refer to a subset of its vertices $S \subset \mathcal{S}$ as a planar separator if the removal of these vertices partitions the graph into two disjoint subgraphs. We call S a vertical planar separator, if $|S| = \overline{h}$ and $\chi^{\mathrm{h}}(S) = \{1, ..., \overline{h}\}$, with $\overline{h}$ denoting the number of different horizontal positions $\{1, ..., \overline{h}\}$ in H, and $\chi^{\mathrm{h}}(S)$ being a function that denotes the set of horizontal positions for each vertex in S.

This algorithm has a computational complexity of $O(nh7^h)$, where n represents the number of customers and h denotes the number of horizontal lines in the Hanan graph $H(\mathcal{V})$.

3.2. A bidirectional layered graph algorithm for a single picklist

In the following, we develop a bidirectional layered graph algorithm for a specific case of the GOPP with a single picklist and a single drop-off point. We first sketch the general idea of our algorithm in Section 3.2.1, before we detail its main components in Section 3.2.2, and show in Section 3.2.3 how the algorithm's performance can be further enhanced via an improved graph representation.

3.2.1. Algorithm sketch: Our algorithmic idea is based on four fundamental steps, which we outline in the following, before we formalize our algorithm.

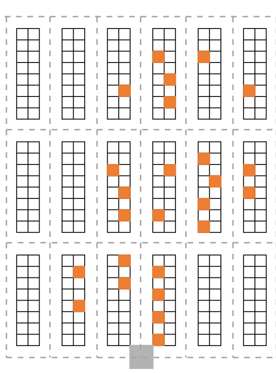
i) We first introduce a compact graph representation for our physical planning problem: Figure 4a depicts a picking zone layout with a single drop-off point and one picklist. Here, the drop-off point is the gray rectangle, and the shelves that must be visited are highlighted in orange. Figure 4b shows its corresponding graph representation. We define a Hanan graph $H(\mathcal{C}) = (\mathcal{C}, \mathcal{E})$ based on the intersections $\mathcal{C}$ of the picking zone and note that its edges $\mathcal{E}$ span all storage positions.

ii) We then note that the transitions defined by Ratliff and Rosenthal (1983) still hold in our setting so that we can use these transitions to describe a picking tour in $H(\mathcal{C})$.

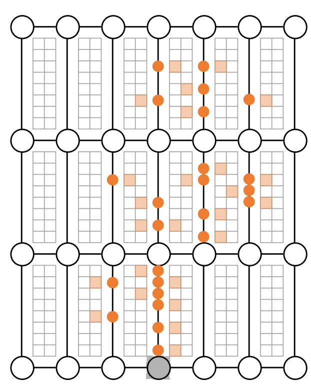
iii) Then, our algorithm's main rationale is to solve a modified Steiner TSP in $H(\mathcal{C})$. This Steiner TSP remains modified from its original definition since instead of a set of terminal vertices, a set of terminal edges $\mathcal{E}^{\mathrm{m}} \subseteq \mathcal{E}$ and the depot vertex must be covered by the tour in its solution. These edges represent sub-aisles that the picker has to visit to pick all items on her list. Here, we use the transitions mentioned above to explore edge traversals and introduce the definition of a *path subgraph* to describe a solution of this Steiner TSP.

iv) In this setting, we use a planar separator to split a path subgraph into *partial path subgraphs.* This allows us to design a problem specific LGA algorithm to solve the modified rectilinear Steiner TSP and thus the GOPP with a single picklist, a single travel mode, and a single drop-off point.

Definition 5 (path subgraph). A subgraph $T = (\mathcal{V}^T, \mathcal{A}^T)$ of G is a path subgraph if it is connected and its degree $\deg v > 0$, and $\deg v \mod 2 = 0$ for $v \in \mathcal{V}^T$.



(a) Physical layout.



(b) Graph representation.

Figure 4 Example of a storage layout.

Definition 6 (partial path subgraph). Let $T = (\mathcal{V}^T, \mathcal{E}^T)$ be a subgraph of $L = (\mathcal{V}^L, \mathcal{E}^L)$ which is a subgraph of $G = (\mathcal{V}^G, \mathcal{E}^G)$. Then, T is an L partial path subgraph of G if there exists at least one subgraph $F = (\mathcal{V}^F, \mathcal{E}^F)$ with $\mathcal{E}^F \subset \mathcal{E}^G \setminus \mathcal{E}^L$ so that T and F form a path subgraph in G.

Compared to a standard LGA implementation as in Cambazard and Catusse (2018), we make the following extensions to derive an algorithm suitable to solve the GOPP with a single picklist and depot: as indicated above, we introduce $\mathcal{E}^{\mathrm{m}} \subseteq \mathcal{E}$ as the subset of edges that must be visited, from here on referred to as mandatory edges, to ensure that we visit all pick positions where items must be picked. Moreover, we develop a bidirectional algorithm that splits $H(\mathcal{C})$ into two subgraphs, explores these subgraphs individually, and then merges the partial solutions to a complete solution.

Although the worst-case complexity of bidirectional algorithms often remains the same as that of their monodirectional counterparts or even gets worse due to an additional merge procedure, empirical evidence for various problems showed superior performance of bidirectional approaches, e.g., in branch-and-price algorithms. Assuming a computationally efficient merge procedure, a bidirectional algorithm does not perform worse than a monodirectional algorithm. It may even be better because exploring the solution space from different directions in two separate searches may reduce the number of states that must be explored, partially by detecting additional dominance relations. As our merge procedure remains computationally efficient (see Appendix E), it seems promising to develop a bidirectional search. While this algorithm was developed for the picking problem, it also appears to be the first that can readily be adapted to the rectilinear Steiner TSP.

3.2.2. Algorithmic Components: To formalize our algorithm, we first label vertical positions v and horizontal positions h in $H(\mathcal{C}) = (\mathcal{C}, \mathcal{E})$. Here, $h \in \{1, \dots, \overline{h}\}$ refers to the h^{th} horizontal line, that is, in our specific setting, to the h^{th} cross-aisle. We label horizontal lines from the lower left corner ($h = 1$) to the upper left corner ($h = \overline{h}$). Analogously, $v \in \{1, \dots, \overline{v}\}$ refers to the v^{th} vertical line, that is, in our specific setting, to the v^{th} aisle. We label vertical lines from the lower left corner ($v = 1$) to the lower right corner ($v = \overline{v}$). We then uniquely identify a vertex $c \in \mathcal{C}$ by its location on the h^{th} horizontal and the v^{th} vertical line of $H(\mathcal{C}) = (\mathcal{C}, \mathcal{E})$. Let $\chi^{\mathrm{h}}(c)$ denote the horizontal position of a vertex with $\chi^{\mathrm{h}}(c) \in \{1, ..., \overline{h}\}$ and let $\chi^{\mathrm{v}}(c)$ denote the vertical position of a vertex c with $\chi^{\mathrm{v}}(c) \in \{1, ..., \overline{v}\}$. We use both functions for single vertices as defined above but also for sets of vertices, then returning a set of positions.

With this notation, we define a *central planar separator* $\hat{S}$, which allows to define two subgraphs B and U of $H(\mathcal{C})$ as shown in Figure 5. Here, B consists of all vertices in $\hat{S}$ and to the left of $\hat{S}$ and its corresponding edges, whereas U consists of all vertices in $\hat{S}$ and to the right of $\hat{S}$ and its corresponding edges. In each of these subgraphs, we define additional subgraphs B_{hv} / U_{hv}, where the edges of B_{hv} are induced by all vertices $c \in B$ for which $(\chi^{\mathrm{v}}(c) < v) \vee (\chi^{\mathrm{v}}(c) = v \wedge \chi^{\mathrm{h}}(c) \leq h)$

holds, and the edges of U_{hv} are induced by all vertices $c \in U$ for which $(\chi^{\mathrm{v}}(c) > v) \vee (\chi^{\mathrm{v}}(c) = v \wedge \chi^{\mathrm{h}}(c) \geq h)$ holds. Then, let R_{hv} / L_{hv} be the planar separators that define B_{hv} / U_{hv} in B / U. Figure 6 shows an example of a subgraph B_{34} with its separator R_{34} in B and a subgraph U_{46} with its separator L_{46} in U.

Definition 7 (central planar separator). We refer to a planar separator S of a Hanan graph $H(\mathcal{U}) = (\mathcal{S}, \mathcal{E})$ as central, if:

1. S is a vertical planar separator according to Definition 10.
2. For all $c_i, c_j \in S$ it holds that $\chi^{\mathrm{v}}(c_j) \leq \chi^{\mathrm{v}}(c_i)$ if $\chi^{\mathrm{h}}(c_j) > \chi^{\mathrm{h}}(c_i)$ and $|\chi^{\mathrm{v}}(S)| = 2$.
3. $||\mathcal{E}'| - |\mathcal{E}''||$ is minimal for the resulting subgraphs $H' = (\mathcal{S}', \mathcal{E}')$ and $H'' = (\mathcal{S}'', \mathcal{E}'')$, with H' being induced by all vertices in or to the left of S and H'' being induced by all vertices in or to the right of S.

Recall that Proposition 2 suggests that we can explore B and U independently to find a cost-minimal path subgraph in $H(\mathcal{C})$, i.e., a shortest tour for the Steiner TSP. Then, the main rationale of our algorithm is to propagate partial path subgraphs in B and U in order to complete these to a cost-minimal path subgraph. To propagate these partial path subgraphs, we use a transition set $\mathcal{T}_e$ which denotes all aforementioned possible transitions $t \in \mathcal{T}_e$ for each edge $e = (c, c') \in \mathcal{E}$.

We develop a bidirectional LGA that iteratively explores subgraphs B_{hv}, U_{hv} in B, U to identify partial path subgraphs. As each subgraph B_{hv}/U_{hv} can contain multiple partial path subgraphs, we identify each partial path subgraph by a state α. Each state $\alpha = \{\mathbf{r}, \mathbf{u}\}$ consists of a vector $\mathbf{r} = (r_1, \ldots, r_{\overline{h}})$ which denotes the degree parity of each vertex in R_{hv}/L_{hv}, and of a vector $\mathbf{u} = (u_1, ..., u_{\overline{h}})$ which denotes the *connected component* to which each vertex in R_{hv}/L_{hv} belongs. Here, both the i^{th} entry r_i of $\mathbf{r}$ and the i^{th} entry u_i of $\mathbf{u}$ state the degree parity, respectively the connected component label, of the vertex that lies on the i^{th} horizontal line in R_{hv}/L_{hv}.

Definition 8 (connected component). A partial path subgraph may contain partial paths that are not necessarily connected with each other but may still yield a feasible solution if they are merged with a second partial path subgraph that preserves the connectedness of the overall path. To trace the number of different partial paths, we use a connected component label u_i given by an integer number. Here, u_i denotes the connected component label of the vertex that lies on the i^{th}

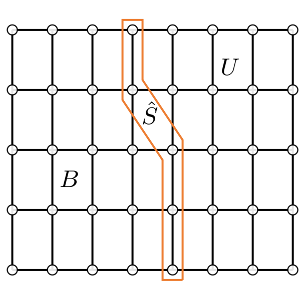


Figure 5 **Example of $\hat{S}$ and its resulting subgraphs.**

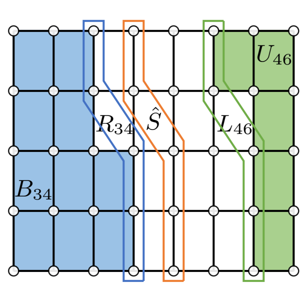


Figure 6 **Example of subgraphs B_{34} and U_{46}.**

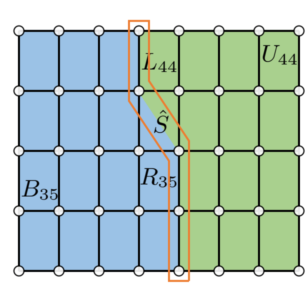


Figure 7 **Fully explored graph with $\hat{S} = L_{44} = R_{35}$.**

horizontal line of $H(\mathcal{C})$ in the planar separator. We initialize the label of the first explored vertex with one and increase the label everytime a transition disconnects c' from c when propagating (c,c'). When a transition reconnects two vertices, both gain the lower connected component label.

Once $L_{hv} = R_{hv} = \hat{S}$ it holds that $B_{hv} = B$ and $U_{hv} = U$, and B and U collectively exploit $H(\mathcal{C})$ exhaustively (Figure 7). We can then identify the optimal path by merging feasible paths from B_{hv}/U_{hv}. To specify our general algorithmic concept, we now define transitions to propagate partial path subgraphs, a feasibility check to either discard infeasible states or to merge partial path subgraphs, and a dominance criterion to discard dominated states.

Transitions: We use the transitions proposed by Ratliff and Rosenthal (1983) as vertical transitions (cf. Figure 3). Doing so straightforwardly yields a transition set that is larger than needed. To derive a minimum transition set and speed up computational times, we separate all edges of set $\mathcal{E}$ of $H(\mathcal{C})$ into a set of mandatory edges $\mathcal{E}^{\mathrm{m}}$ and a set of optional edges $\mathcal{E}^{\mathrm{o}}$. Mandatory edges comprise edges that represent a mandatory sub-aisle while optional edges represent non-mandatory sub-aisles or cross-aisles such that $\mathcal{E} = \mathcal{E}^{\mathrm{m}} \cup \mathcal{E}^{\mathrm{o}}$. Since each edge in $\mathcal{E}^{\mathrm{m}}$ must be visited, transitions *II–VI* are sufficient to propagate mandatory edges. Non-mandatory edges will only be used as connecting edges such that we need only transitions *I–III* to propagate these.

Feasibility Check: We use the information stored in the state $\alpha = \{\mathbf{r}, \mathbf{u}\}$ to evaluate whether a transition leads to an infeasible state or not. Our feasibility check then bases on the degree $r_i \in \mathbf{r}$ and on the connected component label $u_i \in \mathbf{u}$ of each vertex in R_{hv}/L_{hv}. The degree of each vertex $r_i \in \{\mathrm{E}, \mathrm{O}\}$ can be even (E) or odd (O). A partial path subgraph in B_{hv}/U_{hv} may contain multiple partial paths that are not necessarily connected with each other (cf. Definition 6). To trace these, u_i denotes an integer label of the connected component of the partial path ending on the i^{th} horizontal line in R_{hv}/L_{hv}.

Given this information, we can check the feasibility of a transition as follows. A transition leads to an infeasible state if any of the following statements holds:

i) After applying a transition from vertex c to vertex c', c remains with an odd degree but is no longer in R_{hv}/L_{hv}.
ii) After applying a transition, the drop-off point vertex is not connected to the path.
iii) After applying a horizontal transition, a connected component is reduced without a merge.

To check the feasibility during the merge operation, a merge is feasible only if:

i) No vertex in $\hat{S}$ remains with an odd degree.
ii) Only a single connected component label remains.

Dominance Criterion: We use the definition of *equivalence* between partial path subgraphs to develop a dominance criterion. To keep the number of propagated partial path subgraphs as small as possible, we withdraw each partial path subgraph that is equivalent to an already existing partial path subgraph and does not improve its cost. This criterion is based on the rationale that for two *equivalent partial path subgraphs* T and T' in B_{hv} it is sufficient to keep only the one with the lower cost to derive an optimal path subgraph. We establish this equivalence in Proposition 3.

Definition 9 (equivalent partial path subgraphs). We consider a hanan graph $H(\mathcal{C})$ and its subgraphs B and U that collectively cover $H(\mathcal{C})$ exactly. Let T and T' be two distinct partial path subgraphs in B. We call T and T' equivalent if any partial path subgraph F in U that completes T to a path subgraph in $H(\mathcal{C})$, also completes T' to a path subgraph in $H(\mathcal{C})$.

Proposition 3. *We consider a hanan graph $H(\mathcal{C})$ and its subgraphs B and U that collectively cover $H(\mathcal{C})$ exactly. Let T and T' be two distinct partial path subgraphs in B. Then, T and T' are equivalent if their states α and α' are equal.*

Using the described transitions, feasibility check, and dominance check, we propagate partial path subraphs until B_{hv} and U_{hv} collectively exploit $H(\mathcal{C})$ when $B_{hv} = B$ and $U_{hv} = U$. For the algorithmic details, we refer the interested reader to Appendix D.

3.2.3. Improved graph representation: With the modifications described above, we obtain an LGA to solve a picker routing problem for a single picklist, a single drop-off point, and an arbitrary number of aisles and cross-aisles. We now derive an improved graph representation that helps reduce the number of developed labels to enhance the algorithm's computational performance.

The number of labels increases with the number of edges that must be traversed. Hence, we aim to derive a maximal sparse graph that preserves optimality but contains as few edges as possible. We introduce the set $\mathcal{C}^{\mathrm{m}}$ as the set of all vertices that are either adjacent to an edge in $\mathcal{E}^{\mathrm{m}}$ or represent a drop-off point. Then, a graph that preserves the realization of a shortest path between any $c, c' \in \mathcal{C}^{\mathrm{m}}$ preserves optimality. In the following, we provide a reduction technique that improves the sparsity of $H(\mathcal{C})$ while preserving optimality.

Focusing on a planar separator S, we say that this separator is *decreasing* if $\forall\ c, c' \in S,\ \chi^{\mathrm{h}}(c') > \chi^{\mathrm{h}}(c) \ \Rightarrow\ \chi^{\mathrm{v}}(c') \leq \chi^{\mathrm{v}}(c)$ and *increasing* if $\forall\ c, c' \in S,\ \chi^{\mathrm{h}}(c') > \chi^{\mathrm{h}}(c) \ \Rightarrow\ \chi^{\mathrm{v}}(c') \geq \chi^{\mathrm{v}}(c)$. With this notation, we state Proposition 4, which allows to construct a reduced Hanan graph.

Proposition 4. *Let $H' = (\mathcal{C}', \mathcal{E}')$ be a reduced Hanan graph with $\mathcal{E}' \subset \mathcal{E}$ being its edge set, induced by the vertex set $\mathcal{C}' \subset \mathcal{C}$. Then, H' is sparser than $H(\mathcal{C})$, i.e., $|\mathcal{C}'| < |\mathcal{C}|$, $|\mathcal{E}'| < |\mathcal{E}|$ and preserves optimality if we construct $\mathcal{C}'$ as follows.*

1. *We consider two planar separators S^l and S^r, for which the following conditions hold:*
   (a) $|S^l| = |S^r| = h$ *and* $\chi^h(S^l) = \chi^h(S^r) = \{1, ..., \overline{h}\}$,
   (b) *each planar separator S^l, S^r can be split into two partial planar separators with $S^l = S^l_1 \cup S^l_2$ and $S^r = S^r_1 \cup S^r_2$ such that*

i. $\forall\ c \in S_1^l,\ c' \in S_2^l : \chi^h(c) \leq \chi^h(c'),$

ii. $\forall\ c \in S_1^r,\ c' \in S_2^r : \chi^h(c) \leq \chi^h(c'),$

iii. *S_1^l and S_2^r are decreasing, and S_2^l and S_1^r are increasing.*

2. *We can then construct $\mathcal{C}'$ in $O(\overline{h}\overline{v})$ such that $\mathcal{C}^m \subseteq \mathcal{C}'$ and*

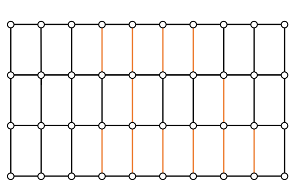
$$\chi^h(c') = \chi^h(c) = \chi^h(c'') \quad \Rightarrow \quad \chi^v(c') \leq \chi^v(c) \leq \chi^v(c'') \quad \forall\ c \in \mathcal{C}',\ c' \in S^l,\ c'' \in S^r.$$

Figure 8 shows an example of such a reduced Hanan graph.

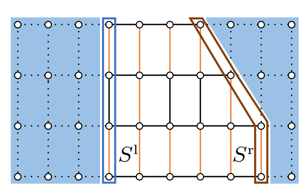
3.3. Extensions for the general GOPP

We now present a general framework that allows to integrate the LGA algorithm of Section 3.2 to solve any variant of the GOPP. The main rationale of this framework leverages a branch-and-prune algorithm to make decisions on i) which subsequent picklists should be processed in parallel, and on ii) the drop-off points at which finished lists should be dropped. To calculate the overall picker walking time for a specific configuration of parallel-processed picklists and drop-off points, we need to solve extended variants of the basic problem studied in Section 3.2. Here, our branch-and-prune algorithm makes use of extended variants of our LGA to solve these subproblems. In the following, we first detail the general branching and pruning framework that embeds our LGA to solve the resulting subproblems in Section 3.3.1. We then discuss characteristics of and extensions that are necessary to integrate *cartless subtours*, a *dynamic batching policy*, and *multiple drop-off points* into our LGA in Section 3.3.2, and derive lower bounds in order to speed-up the branch-and-prune algorithm in Section 3.3.3. We note that we could also use our ILP within the branch-and-prune algorithm and refer to Appendix C for further elaboration.

3.3.1. Branch-and-Prune Algorithm: As a prerequisite to formalize our branch-and-prune algorithm, we first decompose the solution of the GOPP. To this end, we refer to a solution $\sigma = (\vartheta_1, ..., \vartheta_n)$ as an n-tuple of subproblem solutions ϑ_i. Each solution is associated with a cost $C(\sigma) = \sum C(\vartheta_i)$. A subproblem solution represents a single picking tour between two drop-off point visits and is a quadruple $\vartheta_i = (\mathcal{L}_i^{\mathrm{a}}, \mathbf{T}_i, \omega_i^-, \omega_i^+)$ consisting of a set of active picking list labels $\mathcal{L}_i^{\mathrm{a}} \subseteq \mathcal{L}$ which depends on the general set of order list labels $\mathcal{L}$ to be processed, a vector of transitions $\mathbf{T}_i$ that comprises an explicit transition $t_{ij} \in \mathcal{T}_{ij}$ for each edge, a starting drop-off point ω_i^-, and a final drop-off point ω_i^+.



(a) Example instance.



(b) Feasible reduced Hanan graph.

Figure 8 Example of a feasible reduced Hanan graph.

To solve the GOPP using this decomposition, one needs to decide on i) $\mathcal{L}_i^{a*}$, i.e., which picklists are processed in which subproblem, ii) which drop-off points $\omega_i^{+*}, \omega_i^{-*}$ should be used in each subproblem, and on iii) $\mathbf{T}_i^*$, i.e., which transitions form the cost optimal picking tour. To make decision iii) and obtain $\mathbf{T}_i^*$, we leverage an extended variant of our LGA, which we detail in Section 3.3.2. We now develop a branch-and-prune algorithm that allows to make decisions i) and ii), using our extended LGA as an oracle to determine the minimum cost of each subproblem.

We represent our branch-and-bound tree as a graph-theoretic tree, with $\mathcal{N}$ being the set of node labels of the complete tree, 0 being its root node, and $\mathcal{Z}_n \subset \mathcal{N}$ denoting the child nodes of node n. Each node n in this tree represents a subproblem θ_n, identified by a unique triple $\theta_n = (\mathcal{L}_n^a, \omega_n^+, \omega_n^-)$, which may be part of the optimal solution that is given by the cost-minimum path from the tree's root node to a leaf.

Example 2 (branch-and-bound tree). We consider a set of picklists $\mathcal{L} = \{1, 2, 3\}$, a picking zone layout with two drop-off points $a, b \in \mathcal{D}$, and a cart capacity of $\kappa = 2$. Further, the picker is supposed to start her first tour and to end her last tour at a. Figure 9 shows the corresponding fully enumerated branch-and-bound tree, which contains subproblems with four different active picklist configurations $\mathcal{L}_i^a \in \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}\}$, and depot configurations $(\omega_i^-, \omega_i^+) \in \{(a,b), (b,a), (a,a), (b,b)\}$. As one can see, the leaving drop-off point in a child subproblem always remains the entering drop-off point in its direct parent subproblem. This ensures that the subproblems capture the total picker walking distance. Moreover, a path's depth from the root node to a leaf may vary, depending on the configuration of active order lists in each subproblem, because a path up to a leaf must always contain all picklists to represent a feasible solution.

We introduce the following additional notation to detail the algorithm that allows us to explore such a tree efficiently. We associate each node n with a subproblem sequence $\Theta_n = (\theta_i, ..., \theta_n)$, defined by the path from the root node 0 to node n. For example, referring to the example given in Figure 9, the subproblem sequence defined by node $n = 6$ reads $\Theta_6 = (\theta_1, \theta_6)$, while the subproblem sequence for node $n = 16$ is $\Theta_{16} = (\theta_2, \theta_9, \theta_{16})$. Note that the root node is never part of such a

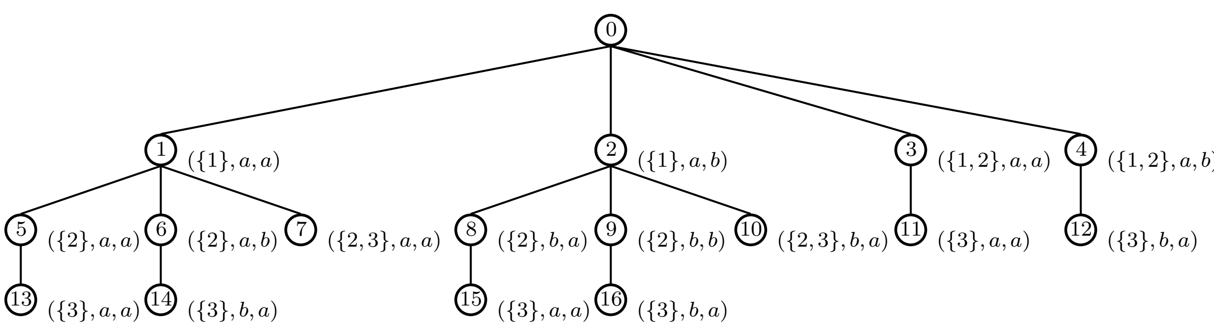


Figure 9 **Branch-and-bound tree representation for $|\mathcal{L}| = 3$, $\kappa = 2$, and $|\mathcal{D}| = 2$.**

sequence as it is not associated with a subproblem, because the first subproblem is not unique due to the varying active picklist configurations.

With ϑ_i being the solution to subproblem θ_i, we can then define the cost C_n of a node n as the sum of the subproblem costs of its corresponding subproblem sequence and note that a subproblem sequence Θ whose subproblems cover all picklists which must be processed constitutes a complete solution σ. We use σ^* to denote the optimal solution and C^* to denote its cost. Further, let $\psi(n)$ denote a function that calculates a lower bound on the cost of a solution that includes the subproblem sequence of n. We refer to Section 3.3.3 for details on the lower bound calculation and note that for leaves of the tree, $\psi(n) = C_n$. To explore the tree, we use sets $\mathcal{N}^{\mathrm{a}}$ and $\mathcal{N}^{\mathrm{s}}$ to keep track of nodes that have not been solved ($\mathcal{N}^{\mathrm{a}}$) and nodes that have already been solved ($\mathcal{N}^{\mathrm{s}}$). Here, $\mathcal{N}^{\mathrm{a}}$ remains an ordered set that can be seen as a priority queue which contains nodes hierarchically ordered by i) the number of processed order lists, starting with the maximum, and ii) nodes with an equal number of processed order lists in increasing order with respect to their lower bound cost. Whenever we insert nodes into $\mathcal{N}^{\mathrm{a}}$, we preserve this order.

Subproblem sequence dominance: Clearly, this branch-and-bound tree grows significantly for larger problem settings depending on the number of picklists and drop-off points. To allow for a tractable exploration of such a tree, we introduce the following dominance relation. We consider two subproblem sequences Θ_n and $\Theta_{n'}$. We then say that Θ_n dominates $\Theta_{n'}$ ($\Theta_n \succ \Theta_{n'}$) if i) the picklists processed in $\Theta_{n'}$ have also been processed in Θ_n, and ii) $C_n + C(\omega_n^+, \omega_{n'}^+) \le C_{n'}$, with $C(\omega_n^+, \omega_{n'}^+)$ being the cost to move from ω_n^+ to $\omega_{n'}^+$. This bound allows to discard a majority of nodes of the branch-and-bound tree early and thus allows to efficiently solve large-size instances.

Pseudo-code: Figure 10 details the pseudo-code of our algorithm. We first initialize $\mathcal{N}^{\mathrm{a}}$ with the children of the root node $\mathcal{Z}_0$ and set C^* to infinity (l.1). We then explore nodes in $\mathcal{N}^{\mathrm{a}}$ in a branch-and-bound fashion. We explore the tree with a depth-first strategy, defined by the order of $\mathcal{N}^{\mathrm{a}}$; in case of a tie, we chose the node with the smaller index (l.3). We then check whether the node is dominated (l.4) or it's lower bound exceeds the so far best-found solution cost (l.5). If any of these checks is true, we discard (prune) the node and continue with the next node in $\mathcal{N}^{\mathrm{a}}$. If none of these checks is true, we either update the optimal solution if the current node is a leaf

1: $\mathcal{N}^{\mathrm{a}} \leftarrow \mathcal{Z}_0, C^* \leftarrow \infty$
2: **while** $\mathcal{N}^{\mathrm{a}} \neq \emptyset$ **do**
3:     $n' \leftarrow getFirstNode(\mathcal{N}^{\mathrm{a}})$
4:     **if** $isNotDominated(n')$ **then**
5:         **if** $\psi(n') < C_{n^*}$ **then**
6:             **if** $isLeafNode(n')$ **then**
7:                 $C^* \leftarrow C_{n'}$
8:                 $\sigma^* \leftarrow \Theta_{n'}$
9:             **else**
10:                 $\mathcal{N}^{\mathrm{s}} \leftarrow \mathcal{N}^{\mathrm{s}} \cup \{n'\}$
11:                 $extend(\mathcal{N}^{\mathrm{a}}, \mathcal{Z}_{n'})$
12: **return** σ^*

Figure 10 Pseudo-code for our branch-and-prune algorithm.

(l.6–8) or add n' to $\mathcal{N}^{\mathrm{s}}$ (l.10) and add its children to $\mathcal{N}^{\mathrm{a}}$ (l.11). We note that the smallest lower bound calculated by $\psi(n)$ is usually not tight, i.e., we do not achieve an exact matching between the smallest lower bound and the global upper bound. However, the lower bound often allows to discard unfavorable unexplored subproblems early.

3.3.2. Subproblem extensions: We recall from Section 3.3.1 that the cost of a subproblem solution results from the cost of its transitions. Accordingly, we can determine the cost of a subproblem by computing a cost-optimal transition vector using the LGA presented in Section 3.2, accounting for the following modifications in settings with cartless subtours, multiple drop-off points, and dynamic batching. We note that all of the detailed modifications are additive, e.g., to solve a subproblem with cartless subtours, multiple drop-off points, and dynamic batching, we apply all modifications outlined below but no further interdependencies between those modifications apply.

Cartless Subtours: Allowing for cartless subtours results in varying travel speeds in sub-aisles, which affect the cost of the transitions. We prove in Proposition 5 that the transitions discussed in Section 3.2 still suffice to calculate an optimal solution. Thus, our algorithm can be applied to problems with cartless subtours without changes.

PROPOSITION 5. *If a picker can switch her travel mode from picking SKUs accompanied by her cart to picking a limited number of items of the same or different SKUs on foot in line with the definition of a cartless subtour, the transition set $\mathcal{T}$ is sufficient to derive an optimal solution.*

Therefore, no changes to our algorithm are necessary to handle cartless subtours. We only add a routine that considers potential cartless subtours and the corresponding travel time reductions when calculating the cost of all transitions during preprocessing.

Multiple drop-off points: We recall that the decision on which drop-off points are used is made in the branch-and-prune phase of our algorithm. This reduces the necessary modifications in our LGA algorithm to account for settings where the starting drop-off point and the final drop-off point are different, i.e., $\omega_i^+ \neq \omega_i^-$. Such a setting leads to optimal subproblem solutions with tours that contain odd-degree vertices, which are prohibited by the feasibility check of our basic LGA, as detailed in Section 3.2. We prove that these odd degrees are always limited to the drop-off point vertices ω_i^+ and ω_i^-, i.e., the start and end of a subproblem's optimal tour.

PROPOSITION 6. *The optimal path subgraph T^* of ϑ_i with $\omega_i^+ \neq \omega_i^-$ comprises exactly two vertices with an odd degree which relate to ω_i^+ and ω_i^-.*

With Proposition 6, we extend the feasibility check of our LGA to a subproblem with $\omega_i^+ \neq \omega_i^-$ by enforcing an odd instead of an even degree for ω_i^+ and ω_i^-.

Dynamic Batching: Under a dynamic batching policy, a picker can pick multiple lists $l \in \mathcal{L}$ at once (as long as the cart's bin capacity κ is not exceeded) and may return to the drop-off point once at least one but not necessarily all picklists are finished to empty one or several bins. As explained in Section 3.3.1, we model such possibilities with subproblem variations: each subproblem comprises the possible transitions to form a picker tour between two drop-off point visits, where the transitions result from the order lists that are active in the respective subproblem.

Specifically, this implies that subproblems exist where the number of active order lists is lower than the bin capacity to reflect cases where a subset of bins is returned to a depot early, although collecting a larger number of picklists in parallel. In these cases, we calculate the respective subproblem's transitions based on the active picklists but consider items of subsequent picklists that are processed in parallel as picked if the resulting transitions cover them. This implicit picking of items from inactive lists reduces the number of items to be picked when their picklist becomes active in a subsequent subproblem, without delaying the drop-off of active picklists. Here, one may achieve even shorter overall picking times by picking more items early from subsequent lists, at the price of delayed drop-offs of active lists. Accounting for this characteristic would require a modified algorithm and remains an interesting future research direction. Here, the decision on which picklists are active in parallel influences the number of consecutive subproblems in a subproblem sequence Θ (see Example 2). We recall that order lists must be processed in chronological order (cf. Section 2.3) and note that this limits the combinations of active picklists.

3.3.3. Lower Bounds for a single subproblem: In the following, we detail the function $\psi(n)$ that we use in Section 3.3.1 to efficiently prune the branch-and-bound tree. Specifically, $\psi(n)$ calculates a lower bound on the cost of a solution that includes the subproblem sequence of n by extending C_n with the maximum of two lower bounds Υ_1 and Υ_2. Here, Υ_1 and Υ_2 are lower bounds for the remaining unexplored subproblems' cost that can be added to the subproblem sequence to get a complete solution. For each unexplored subproblem, we calcuate Υ_1 and Υ_2 as follows.

To calculate Υ_1, we consider our original graph representation $G = (\mathcal{V}, \mathcal{A})$. We derive a minimal spanning tree on a reduced graph $G^c = (\mathcal{V}^c, \mathcal{E}^c)$, where $\mathcal{V}^c$ contains the drop-off vertices and every pick-up position of the subproblem, $\mathcal{E}^c = \{\{i,j\} \mid i \in \mathcal{V}^c, j \in \mathcal{V}^c\}$, and G^c is weighted with the walking time τ_{ij} that results from the Manhattan distance of each arc. Then, a minimal spanning tree is given by $P^c \subseteq \mathcal{E}^c$ with a minimal total weight which yields Υ_1 such that $\Upsilon_1 = \sum_{(i,j)\in P^c} \tau_{ij}$.

To calculate Υ_2, we consider the reduced Hanan graph $H' = (\mathcal{E}', \mathcal{C}')$. We derive a lower bound by summing up i) the cost-minimum transition for each mandatory edge, i.e., $\zeta^*_{ij} = \min_{t \in \mathcal{T}_{ij}} \{\zeta_{ijt}\}$, and ii) costs ζ_{III} for a double traversal of cross-aisles between each pair of aisles with n being the number of aisles remaining in H', such that $\Upsilon_2 = \sum_{(i,j)\in \mathcal{E}^{\mathrm{m}}} \zeta^*_{ij} + \zeta_{III}(n-1)$. For cartless subtours, we treat all transitions of type IV, V, and VI as cartless to calculate Υ_2.

If the leaving and returning depots are not equal, we adapt the calculation of these bounds accordingly, i.e., in this case, the spanning tree for Υ_1 comprises both drop-off vertices, and the minimum cost transition calculation for Υ_2 considers a single transition for both depots.

4. Computational study

The scope of our computational study is twofold. First, we benchmark the basic component of our algorithm against the current state of the art and we also analyze the performance of our algorithmic framework for real-world environments in Section 4.1. Second, we study various warehouse layouts and picking policies based on three case studies from a managerial perspective in Section 4.2.

We conducted all experiments on a standard desktop computer with an Intel Core I7 3.6GHz and 16 GB RAM. Our algorithm was implemented as a single thread code in C++. We solved the ILP using Gurobi 8.1.

4.1. Computational performance

Currently, no algorithm can solve all problem variants considered in this study. The most generic algorithm proposed in parallel to this work by Pansart et al. (2018) focuses on a picker routing problem for a multiblock warehouse limited to a single picklist, a single drop-off point, a static batching, and lacks cartless subtours. We compare the performance of both our monodirectional and our bidirectional LGA, and our ILP (see Appendix C) to the algorithm of Pansart et al. (2018). Since these authors also use dynamic programming, we reimplemented their algorithm in our framework to allow for a fair comparison on the same desktop computer. The benchmark from Theys et al. (2010), which we detail in Appendix G.1 forms the basis for our comparison.

Table 2 compares the performance of our monodirectional (M-LGA) and bidirectional (B-LGA) algorithms, the algorithm of Pansart et al. (2018) (PCC), and our ILP in terms of i) the number of solved instances and ii) the computational times. As can be seen, our LGAs perform much better than the algorithm of Pansart et al. (2018) in terms of both computational times and scalability on large-size instances. In particular, for large-size instances, our LGAs show much better computational times and solve more instances. Comparing our monodirectional algorithm to our bidirectional algorithm, one can see that the bidirectional algorithm shows the best computational times and solves more instances with 11 cross-aisles then the monodirectional algorithm. Note that the computational times for the monodirectional algorithm on instances with 11 cross-aisles and five aisles only appear to be lower since less instances are solved. Accordingly, the comparison is not complete as only solved instances are counted within the average computational times. Our ILP shows longer computational times than all problem specific algorithms but manages to solve a large subset of instances. For the instances with three or six cross-aisles reported as not solved, it still finds an optimal solution but struggles to close the optimality gap within the time limit of an hour.

Remarkably, the ILP solves the most instances with 11 cross-aisles out of all algorithms. While our bidirectional LGA solves 64 instances with 11 cross-aisles, the ILP solves 110 instances with 11 cross-aisles. For most other instances with 11 cross-aisles, the ILP finds a solution but cannot prove its optimality within the given time limit. Only 35 instances remain completely unsolved.

The different performance of these solution approaches can be attributed to specific algorithmic components. Our LGA-based algorithms outperform that of Pansart et al. (2018) for three reasons: first, we use an elaborate online graph reduction which significantly reduces the number of states that must be explored. Notably, the time of this reduction routine is included within the computational times of Table 2. Second, we use improved transition sets which allow for a further reduction in the number of states that must be explored. Third, our feasibility check and equivalence criterion allow us to discard dominated or infeasible states early. In total, the largest share of improvement is due to the graph reduction. Our bidirectional LGA outperforms its mono-directional counterpart as it can discard states earlier by exploring the subgraphs in two directions. While the computational complexity of our LGAs depends on the number of cross-aisles of an instance, the computational complexity of the ILP depends on the total number of arcs. Accordingly, it succeeds in solving a large number of instances with 11 cross-aisles and a small number of aisles but struggles on closing optimality gaps for instances with many aisles, independent of the number of cross-aisles.

In real-world settings, the computational performance of our algorithm can be better or worse than in Experiment 1 due to a multitude of complexity drivers. Hence, we use a second benchmark, detailed in Appendix G.1, to analyze the computational performance of our algorithm with respect to i) the number of cross-aisles, ii) the number of processed picklists and the number of items on each list, and iii) the picking zone layout. Further, we show the impact of our preprocessing technique.

Table 3 shows computational results for a set of benchmark instances with a varying number of cross-aisles. We created 20 instances with a single depot and 50 aisles for each cross-aisle setting,

Table 2 Number of solved instances and computational times in seconds on the Theys et al. (2010) benchmarks.

		Volume based									Random								
Num. of cross-aisles		3			6			11			3			6			11		
Num. of aisles		5	15	60	5	15	60	5	15	60	5	15	60	5	15	60	5	15	60
Instances solved	B-LGA	60	60	60	60	60	60	33	15	2	60	60	60	60	60	60	14	2	-
	M-LGA	60	60	60	60	60	60	27	13	2	60	60	60	60	60	60	9	2	-
	PCC	60	60	60	60	60	60	-	-	-	60	60	60	60	60	60	-	-	-
	ILP	60	60	45	60	38	8	26	10	-	60	60	45	60	49	9	44	30	-
Time	B-LGA	0.004	0.01	0.03	0.16	6.87	45.8	337	555	1320	0.00	0.01	0.04	0.28	10.1	60.1	340	484	-
	M-LGA	0.004	0.01	0.04	0.78	9.42	48.6	153	566	1803	0.00	0.01	0.04	1.39	13.8	63.1	334	1476	-
	PCC	0.006	0.03	0.09	12.4	193	1062	-	-	-	0.01	0.03	0.09	12.9	193	1062	-	-	-
	ILP	0.05	10.4	176	12.2	273	1.17	315	778	-	0.05	3.55	174	2.24	125	1.01	98.9	639	-

The table shows the average computational times of the solved instances on the Theys et al. (2010) benchmark for the algorithm of Pansart et al. (2018) (PCC), our monodirectional LGA (M-LGA), our bidirectional LGA (B-LGA), and our ILP.

Table 3 Computational results for benchmark instance sets with 50 aisles.

Number of cross-aisles	3	4	5	6	7	8
Average computational time [s]	0.02	0.13	1.54	25.6	322	-
Average speed-up factor	1.50	1.66	3.16	6.08	16.4	-
Maximum value speed-up factor	3.07	4.77	20.4	64.5	71.0	-
3rd Quartile speed-up factor	1.46	1.61	2.75	3.87	16.9	-
Median speed-up factor	1.28	1.43	2.09	2.52	6.49	-
1st Quartile speed-up factor	1.20	1.30	1.67	1.92	3.14	-
Minimum value speed-up factor	1.06	1.09	1.19	1.20	1.30	-

The table shows the average computational time and the statistical characteristics of the ratio between the computational time without and with preprocessing for benchmark sets of 20 instances for different numbers of cross-aisles.

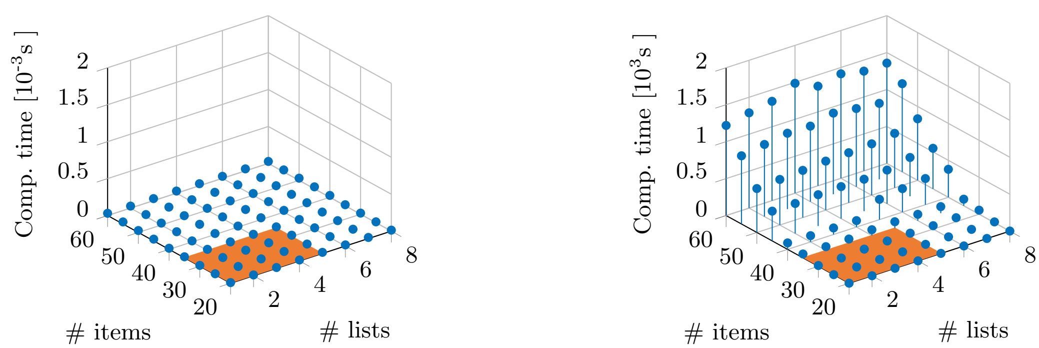
generating pick positions as described in Appendix G.1. We report the average computational time for each instance set. Further we analyze the ratio between the computational times without and with our graph reduction technique, from here on referred to as *speed-up factor*. As can be seen, the overall complexity of the instances increases exponentially with the number of cross-aisles. The computational times remain stable for each instance set, and settings with up to five cross-aisles which are relevant in practice can be solved in less than a few seconds. Instances with seven cross-aisles already take a few minutes of computational time, and instances with eight cross-aisles do no longer show stable computational times as some instances cannot be solved due to memory limitations. Without our graph reduction technique instances with seven cross-aisles already show unstable computational performance. The impact of the speed-up factor varies depending on the instance characteristics. The statistical quantities in Table 3 show that in approximately 50% of the cases our graph reduction technique reduces computational times by at least 50%. In general, the improvement potential of this technique tends to be higher, the higher is the number of cross-aisles.

Besides the number of aisles and cross-aisles, the number of consecutively processed picklists and the number of items on each list influence the computational tractability of our algorithm. Furthermore, this tractability heavily depends on the chosen picking policy and on the number of drop-off points in a picking zone. The simplest real-world configuration comprises a static batching, a single drop-off point, and no cartless subtours on a layout with three cross-aisles and one drop-off point. The most complex real-world configuration comprises a dynamic batching (DS), multiple drop-off points (MD), and cartless subtours (CS) on a layout with three cross-aisles and five drop-off points (cf. Figure 12). Figure 11 shows the computational times for these two extreme configurations. Computational times remain stable below 0.1 second for the simple case. For the complex case, computational times increase significantly for large instances but remain rather insensitive to the number of consecutive picklists due to the dominance criterion introduced in Section 3.3.1. In practice, such algorithms are used in a receding-horizon fashion within a 15-minute interval. This corresponds to a maximum number of four to five consecutive picklists with up to 35 items (see Section 4.2). For these real-world configurations shown by the orange area in Figure 11, we achieve computational times below 90 seconds even for the worst scenario.

Concluding, our algorithm yields both new best known results for existing benchmarks and a generic framework that is amenable to a real-time implementation for a multitude of real-world configurations with respect to picking zone layouts and picking policies. In the following, we extend our experiments by using the bidirectional LGA as it provides the best and a robust performance on instance sizes relevant in practice.

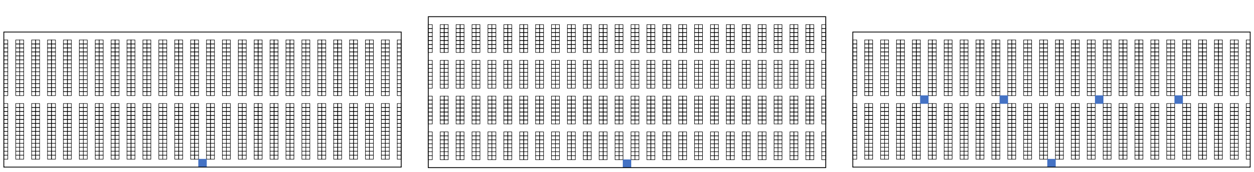
4.2. Impact of different picking policies and picking zone layouts

The three real-world applications discussed in Section 2.2 are similar with respect to the size and throughput of the warehouse. Each comprises roughly 15,000,000 items of SKUs which is the total storage capacity of the Zalando warehouse. This translates into 400,000 items for a single picking zone. A picking zone has a width of 70 to 80 m; each aisle or cross-aisle has a width of 1.5 m; a shelf has a depth of 0.75 m, each storing about 300 items. Hence, 1,400 shelves are necessary to store 400,000 items. Whereas these characteristics are similar for all three warehouses, their layouts vary in terms of the number of cross-aisles and drop-off points: Zalando (Figure 12a) uses the most conventional layout with a single drop-off point (blue square) and three cross-aisles, at the beginning, in the middle, and at the rear of the picking zone; Hermes (Figure 12b) works with a layout having a single drop-off point but uses five cross-aisles to increase the flexibility of the pickers; the Amazon layout (Figure 12c) has three cross-aisles but four additional drop-off points in the middle cross-aisle. In the following, we perform a numerical study based on these case studies to derive insights into the optimal design of picking policies and the respective picking zone layouts.



(a) Layout 1, (SB/SD/NS). (b) Layout 3, (DB/MD/CS).

Figure 11 Computational times dependent on the number of consecutive lists and items per list.



(a) Layout 1 (Zalando). (b) Layout 2 (Hermes). (c) Layout 3 (Amazon).

Figure 12 Picking zone layouts for three real-world cases, by increasing order of complexity.

Experimental Design: To allow for an unbiased analysis, we need to work with identical picklist sequences and picking zone sizes. Hence, we cannot use real data from our three industrial cases but generate realistic synthetic instance data.

To ensure equal picking zone sizes that reflect the spacious conditions of our real-world cases, we choose a picking zone width of 75 m that results in a layout with 25 aisles to cover 50 shelf rows, each row having 28 shelves. The width of a shelf is 2 m, and hence we divide each shelve into two picking positions to derive a distance granularity of 1 m. Further, we generate instances with three and five cross-aisles and one to six drop-off points to account for all picking zone settings from our real-world examples and potential variations.

To ensure identical picklist sequences, we generate realistic picklist sequences as detailed in Appendix G.2 that mimic a potential upstream batching procedure. Here, we account for varying picklist dispersions, i.e., picklists spanning across a varying spatial area within a picking zone, to account for varying batching efficiency. To characterize a picklist dispersion, we use the number of aisles $\beta \in \mathbb{N}$ to the left and right and the share of pick positions $\gamma \in [0, 1]$ above and below a picklist's centroid to define the picking zone area in which the lists items are distributed. Small β, γ values denote clustered picklists, while large β, γ values denote dispersed picklists. The former means that the previous batching planning stage has sucessfully clustered all picking positions in a closely confined area, whereas the latter represents a less successful batch plan where picking positions are widely scattered throughout the picking zone. Since all layouts have the same number of aisles and shelves per aisle, this technique allows us to use the derived picklists on each layout and hence to produce comparable unbiased results.

We perform a full factorial design with respect to the picking policy characteristics (Table 1). To this end, we refer to a picking policy with three identifiers, where the first (SB,DB) indicates whether batching is static (SB) or dynamic (DB), the second (SD,MD) indicates whether a single (SD) or multiple (MD) drop-off points are used, and the third (NS,CS) indicates whether cartless subtours are allowed (CS) or not (NS). Moreover, we consider a varying cart capacity κ between two (Amazon, Zalando) and four (Hermes) bins on a cart that influences the number of parallel processed picklists.

For each resulting setting, we analyze a representative set of 50 instances with five consecutive picklists of 35 items each. This corresponds to a 15-minute planning interval in a receding-horizon setting, which is often used in practice. To evaluate the performance of the different picking policies and picking zone layouts, we analyze the average total picking time reduction (TPTR) related to a picking zone layout with a single depot and the most elementary picking policy (SB/SD/NS). We calculated the TPTR by first comparing the results on a per-instance basis and then computing an average value.

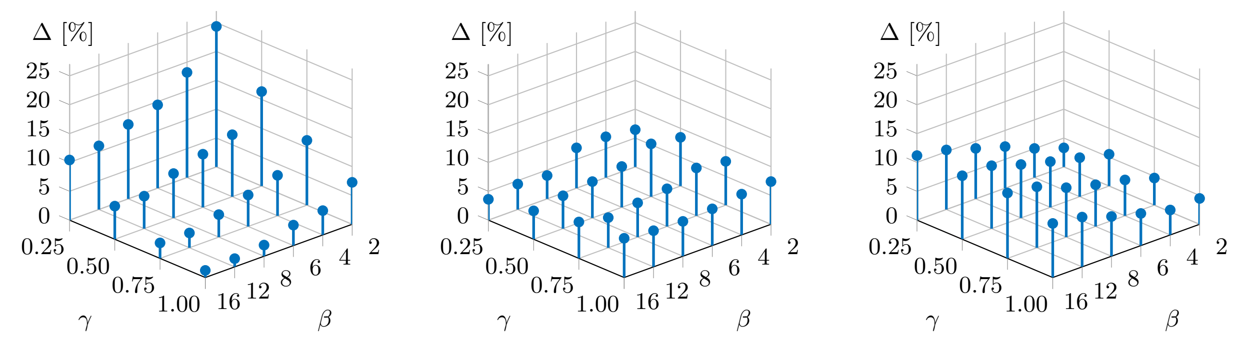
The remainder of this section summarizes our main findings for selected settings, while Appendix I contains detailed results.

Two comments on this experimental design are in order. First, we note that we focus our analyses on the impact of picking policies and the respective picking zone layouts for varying picklist dispersions and cart capacities. We do not analyze the benefit of increasing the cart capacity explicitly as such an analysis would require a different experimental setup that focuses on a larger time horizon. We refer the interested reader to Appendix H for a detailed discussion and prelimary analyses of this impact. Second, although extensive, these studies are not exhaustive since considering layouts with an even number of cross-aisles or further parameter studies may reveal additional benefits. However, we leave these studies for further research as they would require considerable space to allow for rigorous elaboration and discussion.

Picking Policies: Figure 13 focuses on the improvement potential of each picking policy depending on the picklists' dispersion. Here, Figure 13a addresses the impact of multiple drop-off points and shows the average TPTR when using an SB/MD/NS policy with $\kappa = 2$ on a five-cross-aisle layout with six drop-off points. We observe a maximum improvement of 24.3% for clustered picklists with the smallest β, γ. For increasing β, γ, the TPTR may be significantly lower, in the worst case as little as 1%. We see similar trends but different amplitudes for settings with $\kappa \in \{3, 4\}$ and layouts with three cross-aisles. The maximum TPTR for three-cross-aisle layouts remains 3%–4% below the maximum TPTR for five-cross-aisle layouts. Settings with $\kappa \in \{3, 4\}$ show amplitudes that are 7%–8% below the $\kappa = 2$ setting.

Result 1. Multiple drop-off points can yield TPTRs up to 24% on average, given clustered picklists. For dispersed picklists, these TPTRs may become negligible.

Figure 13b explores the impact of dynamic batching and shows the average TPTR when using an DB/SD/NS policy with $\kappa = 4$ on a three-cross-aisle layout with one drop-off point. Here, we observe maximum TPTRs of 9% and minimum TPTRs of 3%. Similar patterns occur for varying



(a) SB/MD/NS TPTR for $\kappa = 2$, 5 cross-aisles, and 6 drop-off points.

(b) DB/SD/NS TPTR for $\kappa = 4$, 3 cross-aisles, and 1 drop-off point.

(c) SB/SD/CS TPTR for $\kappa = 2$, 3 cross-aisles, and 1 drop-off point.

Figure 13 Average TPTR (Δ) for picking policy flexibility levers.

κ and three- as well as five-cross-aisle layouts, with slightly higher amplitudes on three-cross-aisle layouts. While settings with clustered picklists still show greater improvements than settings with dispersed picklists, we do not observe the steep systemic decrease seen for multiple drop-off points. In fact, we observe the highest improvement potential for $\gamma = 0.5$ and $\beta \in \{2, 4\}$.

Result 2. Dynamic batching can yield TPTRs up to 9.2% on average for clustered picklists. For dispersed picklists, the TPTRs may reduce to 3%.

Figure 13c focuses on the impact of cartless subtours and shows the average TPTR when using an SB/SD/CS policy with $\kappa = 2$ on a three-cross-aisle layout with one drop-off point. Interestingly, we observe an inverse pattern with the highest TPTRs being realized for dispersed picklists. Independent of κ or the layout's number of cross-aisles, we observe similar patterns with only slight amplitude variations of roughly 1%.

Result 3. Cartless subtours can yield TPTRs up to 11.3% on average for dispersed picklists. For clustered picklists, the improvement may reduce to 3%.

Figure 14 further details the impact of multiple drop-off points by showing the TPTR for selected levels of picklist dispersion dependent on the number of drop-off points available. As can be seen, the marginal utility of adding an extra drop-off point decreases significantly for settings beyond three drop-off points. This effect is independent of κ or the cross-aisle layout, and we note that the marginal utility decrease is slightly lower for clustered picklists than for dispersed picklists.

Result 4. The deployment of multiple drop-off points shows decreasing marginal utilities for large numbers of drop-off points such that the positive effect of additional drop-off points quickly diminishes the more drop-off points are added.

Figure 15 shows the TPTR when utilizing all picking policy characteristics (DB/MD/CS) for $\kappa = 2$ on a five-cross-aisle layout. We observe a maximum average TPTR of 31% for maximal clustered picklists and decreasing TPTR for more dispersed picklists, yielding a minimum TPTR of 12%. Similar patterns arise for three-cross-aisle layouts and settings with varying κ. While

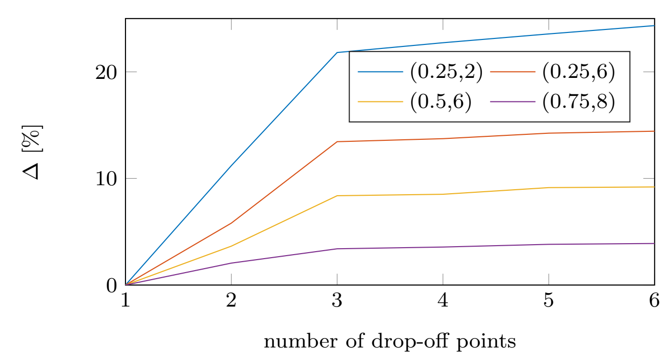


Figure 14 Average SB/MD/NS TPTR for $\kappa = 2$ with 5 cross-aisles for different levels of picklist dispersion (β,γ) dependent on the number of drop-off points.

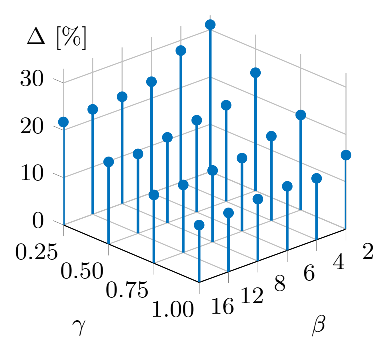


Figure 15 Average DB/MD/CS TPTR with $\kappa = 2$ with 5 cross-aisles and 6 drop-off points.

minimum average improvements remain stable between 11% and 12% for all settings, maximum TPTRs vary between 31% and 21%.

Result 5. Exploiting all picking policy flexibility levers can yield TPTRs up to 31% for clustered picklists. For dispersed picklists, the TPTR may reduce to 11%.

Synthesis: Our observations allow us to synthesize the following insights. First, we note that the TPTR potential of a specific picking policy depends highly on the picklist dispersion, i.e, the efficincy of an upstream batching: the more efficient the upstream batching, i.e., the more clustered the picklists, the larger is the benefit of multiple drop-off points. This allows for a significantly improved routing and early dropping-off of asynchronously finished picklists. For dispersed picklists, it is less likely that the completion time of picklists hugely vary, i.e, that picklists can be dropped off early, and the benefit of multiple drop-offs decreases significantly for inefficient upstream batching. Independent of the batching efficiency, we observe decreasing marginal utilities when increasing the number of deployed drop-off points beyond three drop-off points.

Second, we observe a reversed effect for cartless subtours, which yield the greatest TPTRs in settings with inefficient batching. In these cases, the picking routes span over a large part of the picking zone, which naturally leads to more slopes and subtours that allow for savings, whereas clustered picklists reduce the gains of cartless subtours.

Third, the TPTR potential of dynamic batching appears to be relatively insensitive to the picklist dispersion as changing the order of active picklists bears a TPTR potential in various settings.

Focusing on the overall TPTR potential, the impact of the different levers is additive, i.e., by utilizing all three levers, we reach a TPTR potential similar to the sum of the single TPTR potentials. We do not observe distinct reinforcing effects; across all settings, the overall TPTR potential and the sum of the single TPTR potentials show positive or negative deviations below 2%.

Picking zone layout: Besides the picking policies, changing the picking zone layout with respect to the number of cross-aisles may impact the picking performance. Table 4 shows for all levels of picklist dispersion the range of average TPTRs over all picking policies and cart capacities κ when switching from a three-cross-aisle layout to a five-cross-aisle layout. We observe greater reductions for dispersed picklists with a maximum reduction of 23.2% for $\beta = 16$ and $\gamma = 0.75$. For clustered

Table 4 **TPTR between five and three-cross-aisle layouts for varying picklist distributions.**

β	2				4				6				8				12				16			
γ	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00
max	6.3	6.4	6.3	-0.7	10.5	8.3	10.3	3.3	14.9	10.6	14.2	7.9	15.1	14.1	16.9	9.7	17.5	17.0	21.5	14.1	18.2	17.8	23.2	16.0
Q3	4.7	5.4	4.6	-2.3	8.8	6.8	8.5	1.5	12.3	9.0	12.3	6.4	12.9	11.6	15.6	8.9	14.9	14.6	19.7	12.9	15.6	15.9	20.6	14.2
Q2	4.0	4.3	3.5	-2.9	7.2	5.8	6.9	0.6	11.0	8.2	10.3	3.9	11.9	10.4	13.1	6.0	13.6	13.8	17.8	10.1	15.0	15.4	19.7	12.8
Q1	2.2	3.1	2.8	-4.1	6.1	4.7	6.2	-0.5	10.4	7.6	9.5	2.8	11.3	9.0	12.7	5.0	12.8	12.5	16.3	8.7	13.8	14.3	18.5	10.6
min	0.3	2.1	0.7	-5.1	4.5	3.2	5.0	-1.9	9.1	6.6	7.4	1.8	9.8	7.7	11.7	3.9	11.2	12.1	15.5	7.4	12.4	13.3	17.0	9.1
mean	3.6	4.2	3.6	-3.1	7.5	5.8	7.3	0.6	11.4	8.4	10.8	4.3	12.1	10.5	13.9	6.6	13.9	13.9	18.0	10.5	14.9	15.3	19.7	12.6

This table characterizes the TPTR distribution when switching from a three-cross-aisle to a five-cross-aisle layout.

picklists, we observe significantly lower reductions. For high γ and low β, we even observe increased total picking times in some settings. This effect highlights that additional cross-aisles increase the flexibility of the picker to shortcut into adjacent aisles, but add unused space to the warehouse and increase the lengths of the picking aisles. Particularly for clustered picklists which require only a few aisles to be traversed, additional cross-aisles constitute a significant share of the overall distance, which may worsen the picking time. Hence, there exists a general trade-off between different design and operational objectives, and we cannot conclude that more cross-aisles automatically increase picking performance. This trade-off has been explored in great detail for traditional (single-depot) settings, e.g., by Roodbergen and De Koster (2001) as well as Roodbergen et al. (2008).

Result 6. Increasing the number of cross-aisles significantly reduces the picking time for settings with dispersed picklists. For clustered picklist the reduction is much smaller and increasing the number of cross-aisles may even lead to a loss of picking performance.

5. Conclusions

With the rapid growth in e-commerce, picker routing and its associated policies have become crucial determinants of profitability and competitiveness in warehouse operations. We have presented the first generic and exact algorithmic framework for the GOPP, which constitutes a new state of the art for several known (sub)problem variants and for the generic case. Our algorithm is scalable and amenable to real-time applications since it requires computational times of less than two seconds for realistic warehouse layouts and picking policies. Using this algorithm, we have analyzed the impact of several picking policies and warehouse layouts on the overall performance of a picker.

Although not exhaustive in all design aspects of picking zones, our results allow the derivation of several meaningful managerial insights, identifying trends that may open the field for further research. We have shown that employing enhanced picking policies may reduce the average total picker walking time by up to 20–30% depending on the picking zone layout and cart capacity used, which provides the potential to increase overall flexibility or throughput. Utilizing multiple drop-off points is highly beneficial in case of clustered picklists that result from an efficient upstream batching procedure. Its impact becomes negligible for scattered picklists, i.e., in the absence of an efficient upstream batching. Conversely, cartless subtours appear to be particularly beneficial in the presence of scattered picklists and may partly compensate for inefficient upstream batching, but show only little impact for settings with clustered picklists. Dynamic batching allows for improvements independently of the upstream batching's efficiency and the resulting picklist dispersion. Increasing the number of cross-aisles in a picking zone layout can improve or deteriorate the picking times since deciding on the number of cross-aisles amounts to making a trade-off between routing flexibility and walking distances. Accordingly, using layouts with larger numbers of cross-aisles appears to be particularly beneficial for dispersed picklists but may worsen picking times when an efficient upstream batching generates significantly clustered picklists.

Three final comments in relation with these insights are in order. First, the impact of cartless subtours may be significantly higher or lower, based on the ratio between a picker's speed with and without a cart. The picker speeds used in this work are often observed in practice, which means that our results provide a good baseline, but they should be reexamined if achievable speeds differ significantly. Second, explicitly analyzing the benefit of varying cart capacities may reveal additional interesting insights. Here, we want to emphasize that our results constitute only a first step into this field of research. Extending our framework to provide in-depth analyses of different objectives or integrated batching and order picking, which has been so far mostly studied for rather theoretical warehouse settings, may yield interesting insights for both academics and practitioners.

References

Azadeh K, De Koster RMB, Roy D (2018) Robotized and automated warehouse systems: Review and recent developments. *Transportation Science* 53(4):917–940.

Boysen N, De Koster RBM, Weidinger F (2019a) Warehousing in the e-commerce era: A survey. *European Journal of Operational Research* 277(2):396–411.

Boysen N, Fedtke S, Weidinger F (2018) Optimizing automated sorting in warehouses: The minimum order spread sequencing problem. *European Journal of Operational Research* 270(1):386–400.

Boysen N, Stephan K, Weidinger F (2019b) Manual order consolidation with put walls: The batched order bin sequencing problem. *EURO Journal on Transportation and Logistics* 8(2):169–193.

Bozer YA, Kile JW (2008) Order batching in walk-and-pick order picking systems. *International Journal of Production Research* 46(7):1887–1909.

Cambazard H, Catusse N (2018) Fixed-parameter algorithms for rectilinear Steiner tree and rectilinear traveling salesman problem in the plane. *European Journal of Operational Research* 270(2):419–429.

Chen F, Wang H, Qi C, Xie Y (2013) An ant colony optimization routing algorithm for two order pickers with congestion consideration. *Computers & Industrial Engineering* 66(1):77–85.

Cui R, Zhang DJ, Bassamboo A (2019) Learning from inventory availability information: Evidence from field experiments on Amazon. *Management Science.* 65(3):1216–1235.

Daniels RL, Rummel JL, Schantz R (1998) A model for warehouse order picking. *European Journal of Operational Research* 105(1):1–17.

De Koster RBM, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order picking: A literature review. *European Journal of Operational Research* 182(2):481–501.

De Koster RBM, Van der Poort ES (1998) Routing orderpickers in a warehouse: A comparison between optimal and heuristic solutions. *IIE Transactions* 30(5):469–480.

Gallien J, Weber T (2010) To wave or not to wave? Order release policies for warehouses with an automated sorter. *Manufacturing & Service Operations Management* 12(4):642–662.

Goetschalckx M, Ratliff HD (1988) An efficient algorithm to cluster order picking items in a wide aisle. *Engineering Costs and Production Economics* 13(4):263–271.

Gue KR, Meller RD, Skufca JD (2006) The effects of pick density on order picking areas with narrow aisles. *IIE Transactions* 38(10):859–868.

Hall RW (1993) Distance approximations for routing manual pickers in a warehouse. *IIE Transactions* 25(4):76–87.

Hanan M (1966) On Steiner's problem with rectilinear distance. *SIAM Journal on Applied Mathematics* 14(2):255–265.

Henn S, Wäscher G (2012) Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research* 222(3):484–494.

Hong S, Johnson AL, Peters BA (2012a) Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research* 221(3):557–570.

Hong S, Johnson AL, Peters BA (2012b) Large-scale order batching in parallel-aisle picking systems. *IIE Transactions* 44(2):88–106.

Pansart L, Catusse N, Cambazard H (2018) Exact algorithms for the picking problem. *Computers & Operations Research.* 100:117–127.

Petersen CG (1997) An evaluation of order picking routeing policies. *International Journal of Operations & Production Management* 17(11):1098–1111.

Ratliff HD, Rosenthal AS (1983) Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research* 31(3):507–521.

Roodbergen KJ, De Koster RBM (2001) Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research* 133(1):32–43.

Roodbergen KJ, Sharp GP, Vis IFA (2008) Designing the layout structure of manual order picking areas in warehouses. *IIE Transactions* 40(11):1032–1045.

Statista (2017) Annual retail e-commerce sales growth worldwide from 2014 to 2021. https://www.statista.com/statistics/288487/forecast-of-global-b2c-e-commerce-growt/.

Theys C, Bräysy O, Dullaert W, Raa B (2010) Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research* 200(3):755–763.

Weidinger F (2018) Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research* 95:139–150.

Weidinger F, Boysen N (2018) Scattered storage: How to distribute stock keeping units all around a mixed-shelves warehouse. *Transportation Science.* 52(6):1412–1427.

Weidinger F, Boysen N, Schneider M (2019) Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research* 274(2):501–515.

Zulj I, Kramer S, Schneider M (2018) A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem. *European Journal of Operational Research* 264(2):653–664.

Acknowledgments

We like to thank three anonymous referees, an anonymous associate editor, and the department editor Chung Piaw Teo for their valuable feedback, which helped to improve this paper significantly.

Appendix A: Proofs of propositions

Proof of Proposition 1 [§3.1] See Ratliff and Rosenthal (1983). □

Proof of Proposition 2 [§3.2] See Cambazard and Catusse (2018). □

Proof of Proposition 3 [§3.2] We consider a Hanan graph H and its subgraphs B and U, defined by a vertical planar separator S, that collectively exploit H. Let T and T' be partial path subgraphs in B and let α and α' be their respective states. We then show that $\alpha = \alpha'$ implies equivalence between T and T' by showing that for any partial path subgraph F in U it holds that if T and F form a path subgraph P in H, T' and F also form a path subgraph P' in H.

We note that formally, a subgraph P is a path subgraph if for every vertex c in P it holds that i) $\deg c \bmod 2 = 0$ and ii) $\deg c > 0$, and iii) P is connected.

We now note that by construction conditions i) and ii) hold for any vertex in T and T' that are not contained in S. By definition, conditions i) and ii) also hold for all nodes in $F \setminus S$ and for all vertices in S in case T and F form a path subgraph P. Then, $\alpha = \alpha'$ implies that $r_i = r'_i \; \forall i \in \{1, \dots, \overline{h}\}$ and thus conditions i) and ii) also hold for path subgaph P', which is formed by T' and F.

We now focus on condition iii). Here, two cases exist. If T, T' and F are connected, P and P' remain connected and are path subgraphs. If either T and T' or F have more than one connected component, $\alpha = \alpha'$ implies that $u_i = u'_i \; \forall i \in \{1, \dots, \overline{h}\}$ such that either both P and P' or none of these subgraphs is a path subgraph.

This concludes the proof. □

Proof of Proposition 4 [§3.2] We first show that requiring $\mathcal{C}^{\mathrm{m}} \subseteq \mathcal{C}'$ in combination with conditions a) and b) of Proposition 4 preserves optimality in H', by proving that the length of a shortest path for any pair of $(c, c'),\; c, c' \in \mathcal{C}$ is the same in H' and H. This only holds if the left and right border of H' are convex, which implies that S^{l} and S^{r} must have a concave shape. By construction, S^{l} and S^{r} remain concave.

We now show how a feasible $\mathcal{C}'$ can be constructed from $\mathcal{C}$ in $O(\overline{h}\overline{v})$. We set $\mathcal{C}' = \mathcal{C}$ and proceed as follows.

1. We first traverse $H(\mathcal{C})$ starting in its lower left corner, moving horizontally from left to right along vertices $c \in \mathcal{C}$ with $\chi^{\mathrm{h}}(c) = 1$. We remove the inspected vertex if i) it neither belongs to a mandatory edge, nor ii) it is a drop-off point. The first time we found constraints i) or ii) violated, we store the vertex's vertical position $\hat{v}$. We then start inspecting vertices $c \in \mathcal{C}$ with $\chi^{\mathrm{h}}(c) = 2$ from left to right and stop either if constraints i) or ii) are violated or if iii) $\chi^{\mathrm{v}}(c) > \hat{v}$. We then repeat this procedure until we inspected vertices with $\chi^{\mathrm{h}}(c) = \overline{h}$ and condition i)–iii) is violated.
2. Afterwards, we repeat this procedure starting in the upper left corner of $H(\mathcal{C})$, moving horizontally from left to right and subsequently from top to bottom.
3. We then proceed analogously for the lower and upper right corners of $H(\mathcal{C})$, moving horizontally from right to left and subsequently in the respective vertical direction.

Removing vertices in this order preserves the convexity of $\mathcal{C}'$, which concludes the proof. □

Proof of Proposition 5. [§3.4.1] Recall from Section 2.3 that a cartless subtour starts and ends at the same position and occurs only within a sub-aisle. Accordingly, only circular subtours with a limited number of stops may arise. We consider a sub-aisle in between vertex $c \in \mathcal{C}$ and vertex $c' \in \mathcal{C}$ and differentiate between two cases, depending on a picker's walking time through this sub-aisle by cart $\tau^{\mathrm{c}}_{c,c'}$ or by foot $\tau^{\mathrm{p}}_{c,c'}$:

If $\tau^{\mathrm{c}}_{c,c'} \leq \tau^{\mathrm{p}}_{c,c'}$, partial tours without cart become superfluous.

If $\tau^{\mathrm{c}}_{c,c'} > \tau^{\mathrm{p}}_{c,c'}$, we can still use the standard transitions based on the following reasoning. We note that:

i) Transition *I* disconnects the path, such that it is valid independent of the travel mode.

ii) Transition *II* does not allow to form a slope within a sub-aisle and thus, does not model a cartless subtour in any case.

iii) Transition *III* allows to go back and forth between two vertices but is always dominated by Transitions *IV*, *V*, and *VI*, because a cartless subtour by definition is limited to a sub-aisle.

iv) Transitions *IV* and *V* capture the case that all items are collected with a slope that starts at one end of the sub-aisle. In this case, one can reduce the cost of the transitions by the eligible cartless subtours within the respective slope.

v) The cost of Transition *VI* in the case of cartless subtours can be calculated by considering the cost reduction in eligible subtours for each potential split, identifying its minimum. This minimum may still not be unique. Then, the same argument as for the standard transition holds.

For transitions IV to VI, a slope may contain more items than a picker may pick on a cartless subtour. In this case, we consider only a part of the slope as cartless such that the respective capacity constraint is not violated. Accordingly, one can recalculate the influence of cartless subtours on each transition and its eligible slopes a-priori, such that the same transition set can still be used to run the *layered graph algorithm* (LGA) or the integer linear program (ILP).

This concludes the proof. □

Proof of Proposition 6. [§3.4.1] With $\omega_i^+ \neq \omega_i^-$, T^* remains either as a straight origin-destination path without cycles or contains one or more cycles. In both cases T^* must remain connected so that the only odd degrees relate to ω_i^+ and ω_i^-. This holds since the traversal of each cycle must start and end in the same vertex since T^* remains an Eulerian cycle. In any infeasible case, additional odd degrees arise. □

Appendix B: NP-hardness proof for the generalized order picking problem (GOPP) with multiple picklists and multiple drop-off points

Here, we prove that the GOPP as defined in Section 2.3 (i.e., with multiple picklists to be assembled according to a given processing sequence, a cart with a capacity of κ orders, and multiple drop-off points) is strongly NP-hard even if each picklist demands just a single item. Note that a straightforward transformation from the TSP is not available for our problem, because picker routing in parallel aisle warehouses is solvable in polynomial time (Ratliff and Rosenthal 1983). Instead, our transformation is from the sorting buffer problem (dubbed SBP), which is NP-hard in the strong sense (Asahiro et al. 2012). The SBP in its feasibility version is defined as follows.

Sorting Buffer Problem: Given a sequence η of n items, each item i is to be colored in color $k(i)$ by a server, which has a random access input buffer for up to g parallel items. According to η, items are moved into the buffer. At any step, a color k is selected and all items in the buffer of color k are removed from the buffer and processed by the server. Freed buffer space is filled by succeeding items of η; if some of these new items also demand color k, they are instantaneously removed until no item in the buffer is of color k. Then, a new color is selected and the process repeats, until no items remain in η. Given no initial color, we have to decide whether a processing plan with at most K color changes exists.

We transform an instance I of SBP into an instance I' of our GOPP in pseudo-polynomial time as follows. For each color k of I, we introduce an aisle within a rectilinear warehouse layout. Each item i with color $k(i)$ of SBR corresponds to a picklist demanding just a single item to be retrieved for a picking position located in the middle of the aisle corresponding to the respective color. Thus, if we have four items of color white in SBP, we introduce four picklists each demanding an item stored at a picking positions in the middle of the 'white aisle' of GOPP. Furthermore, in the middle of each aisle we have a drop-off point. Note that, for a matter of convenience, we assume that all items stored in the same aisle and each aisle's drop-off point are located in the middle of the aisle and have zero distance among each other. This proof can easily be extended by (small) distances among an aisle's picking positions and drop-off point. Additionally, we have the picking cart of GOPP, which resembles the input buffer of SBP. The capacity of the cart equals that of the input buffer, so that we have $\kappa = g$. The sequence η of SBP equals the GOPP sequence of picklists. The length of each aisle is 2λ, so that the distance towards the middle of each aisle, where the picking positions and the drop-off points are located is λ. The distance all along the front cross-aisle in order to reach the parallel picking aisles is ρ. We assume $\lambda > K \cdot \rho$. Initially, the picker is located in the cross aisle and the question we ask is whether there exists a solution to I' with a total picking distance $Z < 2\lambda \cdot K$. Our transformation scheme is depicted in Figure 18.

If I is a yes-instance, visiting the aisles within GOPP in the same sequence as switching colors within SBP leads to a solution to I' with $Z < 2\lambda \cdot K$. When visiting an aisle with the picking cart all picklists with their items stored in this aisle can be completed and handed over at the drop-off point. Since we have no distance among the picking positions of this aisle and the drop-off point, all succeeding picklists put onto the cart according to given sequence η and having their items stored in the current aisle can also be processed without causing additional picker walking. Thus, each aisle visit causes a distance of 2λ to walk back and

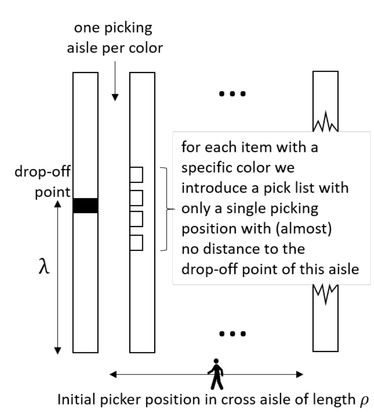


Figure 18 Transformation scheme of complexity proof.

forth the middle of the aisle, except of the last aisle where picking stops after handing over the last picklists. Since even completely passing the front cross aisle (of length ρ) K times does not exceed λ, the resulting solution to I' does not exceed $Z < 2\lambda \cdot K$.

On the other hand, any solution to I' of GOPP with $Z < 2\lambda \cdot K$ is also a feasible solution to I of SBP. Switching colors in the same sequence as changing aisles inevitably leads to K color changes, which completes our complexity proof. □

Appendix C: An integer linear program for a single picklist

Instead of designing an LGA to solve the order picking problem for a single depot and a single picklist, we can also develop a concise and efficient ILP to solve this problem. We define this ILP on a modified, directed graph $G' = (\mathcal{C}', \mathcal{A})$ which results from transforming the reduced graph H' into a directed graph. In addition to the set of all arcs $a = (c, c') \in \mathcal{A}$ with $c, c' \in \mathcal{C}'$, $\mathcal{A}^{\text{m}}$ denotes all mandatory arcs, i.e., arcs corresponding to sub-aisles that must be traversed, and $\mathcal{A}^{\text{d-}}, \mathcal{A}^{\text{d+}}$ denote the subset of arcs leaving or entering the vertex that represents the drop-off point. Further, $\mathcal{A}_i$ denotes for each aisle i, the subset of cross-aisle arcs that are located between the i^{th} and the $(i+1)^{\text{st}}$ aisle. We use the cut sets $\delta^+(a)$ and $\delta^-(a)$ to denote all predecessor and successor arcs of a, and a' to denote the inverse arc of a and define $\overline{\mathcal{A}}^{\text{m}} = \{(c, c') \in \mathcal{A}^{\text{m}} \mid \chi^{\text{v}}(c) < \chi^{\text{v}}(c') \vee \chi^{\text{h}}(c) < \chi^{\text{h}}(c')\}$, which allows us to efficiently model slope transitions on mandatory arcs. We use binaries x_a to indicate whether an arc is traversed $(x_a = 1)$ or not $(x_a = 0)$ and binaries y_{at} to indicate whether an arc is sloped with a specific transition t $(y_{at} = 1)$ or not $(y_{at} = 0)$. Specifically, we cover transitions I, II, III from Figure 3 with the binary traversal variables x_a and $x_{a'}$, while we use the y_{at} variables to model potential slopes with the limited transition set $\mathcal{T}' = \{IV, V, VI\}$. The coefficients ζ_a and ζ_{at} denote the predefined cost for each traversal or slope. Further, we use binary z_{ab} to trace if in a tour, arc b is the direct successor of arc a. Binary o_a identifies the first arc of a tour, while binary d_a identifies the last arc of a tour.

With this notation as summarized in Table 5, our ILP is as follows:

$$\text{minimize} \quad Z = \sum_{a \in \mathcal{A}} \zeta_a x_a + \sum_{a \in \overline{\mathcal{A}}^{\text{m}}, t \in \mathcal{T}'} \zeta_{at} y_{at} \tag{1}$$

subject to

$$x_b \geq \sum_{a \in \delta^+(b)} z_{ab} \qquad b \in \mathcal{A} \tag{2}$$

Table 5 Notation used in the ILP formulation.

$\mathcal{A}$	set of all arcs	ζ_a	cost of traversing arc a
$\mathcal{A}^{\text{m}}$	set of all mandatory arcs	ζ_{at}	cost of sloping arc a with transition t
$\mathcal{A}^{\text{d-}}$	set of all depot leaving arcs	a'	inverted arc of a
$\mathcal{A}^{\text{d+}}$	set of all depot entering arcs		
$\mathcal{A}_i$	cross-aisle arcs between the i^{th} and $(i{+}1)^{\text{st}}$ aisle	x_a	a is traversed $(x_a = 1)$ or not $(x_a = 0)$
$\overline{\mathcal{A}}^{\text{m}}$	mandatory arc subset	y_{at}	a is covered with slope t $(y_{at} = 1)$ or not $(y_{at} = 0)$
$\mathcal{T}'$	limited transition set $(\mathcal{T}' = \{IV, V, VI\})$	z_{ab}	b is traversed after a $(z_{ab} = 1)$ or not $(z_{ab} = 0)$
$\delta^-(a)$	cut set yielding all successor arcs of a	o_a	a is the first arc of the tour $(o_a = 1)$ or not $(o_a = 0)$
$\delta^+(a)$	cut set yielding all predecessor arcs of a	d_a	a is the last arc of the tour $(d_a = 1)$ or not $(d_a = 0)$

$$x_a \geq \sum_{b \in \delta^-(a)} z_{ab} \qquad a \in \mathcal{A} \quad (3)$$

$$\sum_{b \in \delta^-(a)} z_{ab} \geq x_a \qquad a \in \mathcal{A} \setminus \mathcal{A}^{\text{d+}} \quad (4)$$

$$\sum_{b \in \delta^-(a)} z_{ab} + d_a \geq x_a \qquad a \in \mathcal{A}^{\text{d+}} \quad (5)$$

$$\sum_{a \in \delta^+(b)} z_{ab} \geq x_b \qquad b \in \mathcal{A} \setminus \mathcal{A}^{\text{d-}} \quad (6)$$

$$\sum_{a \in \delta^+(b)} z_{ab} + o_b \geq x_b \qquad b \in \mathcal{A}^{\text{d-}} \quad (7)$$

$$x_a + x_{a'} + \sum_{t \in \mathcal{T}'} y_{at} \geq 1 \qquad a \in \overline{\mathcal{A}}^{\text{m}} \quad (8)$$

$$\sum_{a \in \mathcal{A}^{\text{d-}}} o_a = 1 \quad (9)$$

$$x_a \geq o_a \qquad a \in \mathcal{A}^{\text{d-}} \quad (10)$$

$$\sum_{a \in \mathcal{A}^{\text{d+}}} d_a = 1 \quad (11)$$

$$x_a \geq d_a \qquad a \in \mathcal{A}^{\text{d+}} \quad (12)$$

$$\sum_{t \in \{IV, VI\}} y_{at} \leq \sum_{b \in \delta^+(a)} x_b \qquad a \in \overline{\mathcal{A}}^{\text{m}} \quad (13)$$

$$\sum_{t \in \{V, VI\}} y_{at} \leq \sum_{b \in \delta^-(a)} x_b \qquad a \in \overline{\mathcal{A}}^{\text{m}} \quad (14)$$

$$\sum_{a \in \mathcal{A}_i} x_a \geq 2 \qquad i \in \{1, ..., \overline{v} - 1\} \quad (15)$$

$$\sum_{a \in \mathcal{B}} x_a \leq \sum_{a \in \mathcal{B}^-} x_a |\mathcal{A}| \qquad \mathcal{B} \subseteq \mathcal{A} \setminus (\mathcal{A}^{\text{d+}} \cup \mathcal{A}^{\text{d-}}) \quad (16)$$

$$\begin{aligned} &x_a \in \{0,1\} \quad a \in \mathcal{A}; \qquad z_{ab} \in \{0,1\} \quad a, b \in \mathcal{A}; \qquad y_{at} \in \{0,1\} \quad a \in \overline{\mathcal{A}}^{\text{m}}, t \in \mathcal{T}'; \\ &o_a \in \{0,1\} \quad a \in \mathcal{A}^{\text{d-}}; \qquad d_a \in \{0,1\} \quad a \in \mathcal{A}^{\text{d+}}. \end{aligned} \quad (17)$$

Our objective minimizes the total cost, which results from traversing or sloping arcs. Constraints (2) and (3) link arc and sequence binaries, Constraints (4) and (5) ensure that each but the last traversed arc has a successor, and similarly Constraints (6) and (7) enforce the existence of a predecessor for all but the first traversed arc. Constraints (8) ensure that every sub-aisle that must be visited is traversed or sloped. Constraints (9) and (10) uniquely identify the first arc of a tour, while Constraints (11) and (12) uniquely identify the last arc of a tour. Constraints (13) and (14) are connecting slopes and preceding and successive traversals, while Constraints (15) ensure that the picker traverses cross-sub-aisles between two aisles in the reduced graph at least twice. Constraints (16) ensure connectivity with $\mathcal{B}^-$ being the set of all arcs that are successor arcs of arcs in $\mathcal{B}$ whose end-vertex is not contained in the vertex set induced from all arcs in $\mathcal{B}$. Finally, Constraints (17) define the domains of the variables.

Since the number of Constraints (16) grows exponentially with the size of $\mathcal{A}$, we add these in a lazy fashion as feasibility cuts in the solver. By so doing, we obtain an integer problem that remains tractable for many instances of small and medium size. We note that the cost coefficients of the x_a and y_{at} variables can be preprocessed from the transitions in Figure 3. Accordingly, these costs remain equal to the costs of the state transitions used in the bidirectional LGA in Section 3.2 and thus, the bidirectional LGA and the proposed ILP solve the same reduced version of the GOPP and obtain the same optimal value.

One comment on this ILP is in order. We note that one could simplify this ILP further for the prevalent case of a single drop-off point and a single picklist, i.e., one could omit the z_{ab}, o_a, d_a variables and the $\mathcal{A}^{\text{d+}}$, $\mathcal{A}^{\text{d-}}$ sets. We state a more comprehensive ILP formulation as it is applicable to multiple drop-off points without major modifications. Accordingly, one could use the ILP instead of the LGA as an oracle in our branch-and-prune algorithm with minor changes: for cartless subtours and dynamic batching, the changed transition calculations as described for the LGA in Section 3.3 apply. For multiple drop-off points it suffices to consider all depots in the respective depot arc sets $\mathcal{A}^{\text{d+}}$ and $\mathcal{A}^{\text{d-}}$, and to drop Constraints (15) which is a valid inequality for the special case with one depot.

Appendix D: Pseudo-code:

In the following, we detail the pseudo-code for the LGA described in Section 3.2. We propagate partial path subraphs as described in Section 3.2 until B_{hv} and U_{hv} collectively exploit $H(\mathcal{C})$ when $B_{hv} = B$ and $U_{hv} = U$. Here, we use a layered graph structure to store the states of partial path subgraphs in sets A_j^{B} / A_j^{U}, which denote the states α that correspond to partial path subgraphs in the j^{th} B_{hv} / U_{hv} subgraph of B / U. Then, we obtain the optimal picking path by merging the states on the last layers A_j^{B} and A_j^{U} that contain the states of partial path subgraphs explored in B and U. We elaborate the merge procedure in Appendix D and detail only the forward and backward propagation in the following.

To keep the pseudo-codes of the corresponding algorithms concise, we first introduce the ordered sets $\mathcal{E}^{\text{B}}$ and $\mathcal{E}^{\text{U}}$, and denote the cost (i.e., the picking time) of a partial path subgraph associated with a state α as $C(\alpha)$. Let $\mathcal{E}^{\text{B}}$ contain all edges of B, ordered such that one traverses B starting at its lower-left corner, processing vertical edges first and horizontal edges second, when processing edges in the order of $\mathcal{E}^{\text{B}}$. Analogously, $\mathcal{E}^{\text{U}}$ contains all edges of U, ordered such that one traverses U starting at its upper right corner, processing vertical edges first and horizontal edges second.

Figure 19 provides the pseudo-code of our forward propagation. We first initialize A_0^{B} with an artificial unconnected state α_0 (l.1). Then, we start at the bottom left corner of B and explore partial path subgraphs by propagating vertical edges (i.e., sub-aisles) first and horizontal edges (i.e., sub-cross-aisles) iteratively (l.2). Here, each state related to a partial path subgraph on the next subgraph results from propagating a state from its preceding subgraph and a transition (l.4–l.6). For each propagated state, we check its feasibility (l.7). If the generated state is feasible, we check whether the current layer of states already contains an equivalent partial path subgraph (l.8). In this case, we keep only the state with the lower cost and discard the other one (l.9 and l.10). Otherwise, we add the generated state to the set of states A_l^{B} of the corresponding layer l, which refers to the l^{th} subgraph (l.11 and l.12).

The backward propagation (Figure 20) works analogously but starts at the upper right corner of U, propagating vertical edges first and horizontal edges consecutively. To do so, we use the ordered set $\mathcal{E}^{\text{U}}$ instead of $\mathcal{E}^{\text{B}}$.

We show in Appendix F, that both algorithms have a worst-case complexity of $O(\overline{v}\overline{h}7^{\overline{h}})$.

```
A_0^B ← {α_0}, picklist ← 0
for e ∈ E^B do
    picklist ← picklist + 1
    for α ∈ A^B_{picklist−1} do
        for t ∈ T_e do
            α′ ← propagate(t, α)
            if feasible(α′) then
                if ∃ α′′ ∈ A^B_{picklist} : α′′ == α′ then
                    if C(α′) < C(α′′) then
                        α′′ ← α′
                else
                    A^B_{picklist} ← A^B_{picklist} ∪ {α′}
```

Figure 19 Pseudo-code of the forward propagation.

```
A_0^U ← {α_0}, picklist ← 0
for e ∈ E^U do
    picklist ← picklist + 1
    for α ∈ A^U_{picklist−1} do
        for t ∈ T_e do
            α′ ← propagate(t, α)
            if feasible(α′) then
                if ∃ α′′ ∈ A^U_{picklist} : α′′ == α′ then
                    if C(α′) < C(α′′) then
                        α′′ ← α′
                else
                    A^U_{picklist} ← A^U_j ∪ {α′}
```

Figure 20 Pseudo-code of the backward propagation.

Appendix E: Merge procedure

In the following, we detail the merge procedure used in our bidirectional LGA implementation. Given the final-layer sets A^{B} and A^{U} that contain the states of partial path subgraphs explored in B and U (see Section 3.2.3), we can identify a cost-optimal path subgraph as

$$\underset{\alpha \in A^{\mathrm{B}}, \alpha' \in A^{\mathrm{U}}}{\arg\min} \; C(\alpha) + C(\alpha') \tag{18}$$

such that α and α' can be merged to a feasible path subgraph. Here, we can check feasibility in $O(\overline{h})$ by i) iteratively merging the connected component labels of α and α' to ensure that merging both partial path subgraphs yields a subgraph with a single connected component, and ii) ensuring that $\mathbf{r}^{\alpha} = \mathbf{r}^{\alpha'}$. Then, solving (18) straightforwardly via pairwise comparison results in $O(|A^{\mathrm{B}}||A^{\mathrm{U}}|)$ necessary comparisons, such that the overall computational complexity is $O(\overline{h}|A^{\mathrm{B}}||A^{\mathrm{U}}|)$.

Observation 1. Let $\mu = \sum_{i \in \{1,\ldots,h\}} r_i^{\alpha}$ be the total parity in $\hat{S}$ for a state $\alpha \in A^{\mathrm{B}}$ and $\xi = \sum_{i \in \{1,\ldots,h\}} r_i^{\alpha'}$ be the total parity in $\hat{S}$ for a state $\alpha' \in A^{\mathrm{U}}$. A merge of two partial path subgraphs can only be feasible if μ and ξ are either both even or both odd. Moreover, the sum of parity degrees must be even for each vertex in $\hat{S}$.

To reduce the number of necessary comparisons, we exploit Observation 1 and divide A^{B} and A^{U} into subsets such that $A^{\mathrm{B}} = \bigcup_{i \in \mathcal{I}} A_i^{\mathrm{B}}$ and $A^{\mathrm{U}} = \bigcup_{i \in \mathcal{I}} A_i^{\mathrm{U}}$. We derive $|\mathcal{I}| = 2^{\overline{h}}$ subsets, each containing states with an equal $\mathbf{r}_{\alpha}$ / $\mathbf{r}_{\alpha'}$. Then, we can limit the number of necessary comparisons to $O(\sum_{i \in \mathcal{I}} |A_i^{\mathrm{B}}||A_i^{\mathrm{U}}|)$, as $\mathbf{r}_{\alpha} = \mathbf{r}_{\alpha'}$ must hold for a feasible merge, and obtain a worst-case computational complexity of $O(\overline{h} \sum_{i \in \mathcal{I}} |A_i^{\mathrm{B}}||A_i^{\mathrm{U}}|)$.

To further speed up our merge procedure, we sort the states in each subset in increasing order with respect to their costs. This allows us to efficiently exploit potential merge moves and to exit the comparisons once a feasible solution is found as all succeeding pairs would show a higher potential objective value.

Appendix F: Computational complexity of the forward and backward LGA

In the following, we prove that both the forward and backward LGA as introduced in Section 3.2.3 have a worst-case time complexity of $\mathcal{O}(\overline{v}\overline{h}7^{\overline{h}})$, with $\overline{v}$ being the number of aisles and $\overline{h}$ being the number of cross-aisles of the underlying instance. To keep this paper concise, we discuss the proof for the forward LGA in the following and note that it holds analogously for the backward LGA.

We recall that our algorithm advances through the graph by exploring one arc with each iteration. Accordingly, the number of invocations of its main body (see Figure 8, lines 2–12) is bounded by the total number of arcs, which is in $\mathcal{O}(\overline{h}\overline{v})$. At each iteration, the algorithm propagates all partial path subgraphs that resulted from the previous iteration, herein discarding infeasible or dominated partial path subgraphs. We now show that the number of feasible states is bounded by some function $f \in \mathcal{O}(7^{\overline{h}})$ to prove the proposed worst-case time complexity bound of $\mathcal{O}(\overline{v}\overline{h}7^{\overline{h}})$.

Let $\Omega(\overline{h})$ be the set of feasible and distinct states $\alpha = \{\mathbf{r}, \mathbf{u}\} = \{(r_1, \ldots, r_{\overline{h}}), (u_1, \ldots, u_{\overline{h}})\}$ that result when running our algorithm. Further, let $\Omega'(\overline{h})$ be the set of states where for each $\alpha' = \{(r'_1, \ldots, r'_{\overline{h}}), (u'_1, \ldots, u'_{\overline{h}})\} \in \Omega'(\overline{h})$ the following claims hold:

1. The connected component labels of α' form a non-crossing partition of $\mathbf{u}$ with respect to the ordering induced by vertex indices $[1, \overline{h}]$.
2. Each connected component has an even number of vertices with odd degree.

Cambazard and Catusse (2018) show that $|\Omega'(\overline{h})|$ is bounded by $\mathcal{O}(\frac{(4+\sqrt{8})^{\overline{h}+1}}{\sqrt{(\overline{h}+1)^3}}) = \mathcal{O}(7^{\overline{h}})$ for a general rectilinear Steiner traveling salesman problem (TSP). Accordingly, we prove in the following that for our problem specific transition set $\alpha \in \Omega(\overline{h})$ implies $\alpha \in \Omega'(\overline{h})$. The following two proofs verify Claims 1 and 2 for $\alpha \in \Omega(\overline{h})$, which proves the existence of some $f \in \mathcal{O}(7^{\overline{h}})$ that bounds the number of feasible states in our algorithm and thus verifies $\mathcal{O}(\overline{v}\overline{h}7^{\overline{h}})$ for the algorithm's worst-case time complexity.

Proof of Claim 1. We prove Claim 1 by contradiction. Let $\alpha = \{\mathbf{r}, \mathbf{u}\}$ be a state for which the claim does not hold. Then there exist $a, b, d, e \in [1, \overline{h}]$ with $a < b < d < e$ and $u_a = u_d \neq u_b = u_e$. From $u_a = u_d$ and $u_b = u_e$ it follows that there exists paths π_{ad} and π_{be} from vertices a and b to vertices d and e, respectively in all shortest feasible path subgraphs represented by α. From the construction of $H(\mathcal{C})$, it follows that there exists some c that is both in π_{ad} and π_{be}. Accordingly, there exists a path from vertex a to c and from c to b, which implies that $u_a = u_b$. □

Proof of Claim 2. It is easy to see that Claim 2 holds for the initial state. For any other state, let $\alpha' \in \Omega(h)$ be a feasible state and let $\alpha^t = \{r^t, u^t\} \in \Omega(h)$ denote its successor, such that α^t results from α' by applying transition $t \in \mathcal{T}$ on edge $e = (c_o, c_d) \in \mathcal{E}$ that can be either a horizontal edge, i.e., a sub-cross-aisle $e \in \mathcal{E}^{\mathrm{c}}$, or a vertical edge, i.e., a sub-aisle $e \in \mathcal{E}^{\mathrm{s}}$. We now discuss each transition separately. To keep this discussion concise, we say that a vertex is even (odd) if it has an even (odd) degree.

$t = I$: If $e \in \mathcal{E}^{\mathrm{s}}$, then $\mathbf{r}^t = \mathbf{r}'$ and $\mathbf{u}^t = \mathbf{u}'$. If $e \in \mathcal{E}^{\mathrm{c}}$, then $r^t_i = r'_i \ \forall i \in [1, h] \setminus \{\chi^{\mathrm{h}}(c_d)\}$ and $r'_{\chi^{\mathrm{h}}(c_d)} = 0$.

$t = II$: We first focus on horizontal edges $e \in \mathcal{E}^{\mathrm{c}}$, where we note that the algorithms propagation order always ensures that c_d is unconnected in α but connected in α^t, and consider two cases: i) if c_o is (un-)connected and even, propagating e with transition II is infeasible as c_o remains odd forever; ii) if c_o is connected and odd, by propagating e with transition II the overall number of odd vertices remains unchanged and the claim holds.

We now focus on vertical edges $e \in \mathcal{E}^{\mathrm{s}}$. Here, we consider four cases: i) if c_o and c_d are (un-)connected and even, applying transition II yields an odd degree at both vertices and the claim holds. ii) if c_o is (un-)connected and even, and c_d is connected and odd, the overall number of vertices with an odd degree remains unchanged and the claim holds. iii) if c_o is connected and c_d is (un-)connected and even, the claim holds in analogy to ii) if c_o is odd, and follows by induction if c_o is even as propagating transition II causes c_o to become odd but merges an odd vertex into the connected component. iv) if c_o and c_d are connected c_d must be odd. In case c_o is odd, too, transition II forces both vertices to become even and removes an even number of odd vertices such that the overall number of odd vertices remains even. In case c_o is even in α', it becomes odd in α^t while c_d becomes even such that the overall number of odd vertices remains unchanged.

$t = III$: Transition III does not modify the vertex degree unless a new connected component is added. In the former case the claim clearly holds. In the latter case, both vertices of the new connected component have an even degree and the claim follows.

$t = IV$: $r_i^t = r_i'$ and $u_i^t = u_i'$ for $i \in [1, h]$ unless $r_i^t = 0$. In this case, a new connected component comprising only c_o results. As c_o has an even degree, the claim holds.

$t = V$: Analogous to $t = IV$.

$t = VI$: Follows from cases $t = IV$ and $t = V$.

We conclude that the claim holds for all transitions whenever α^t is feasible.

Then, Claim 2 holds in general. □

Appendix G: Instance generation

This appendix details our instance generation. In Section G.1, we detail the instances and benchmark sets used in the computational performance experiments (cf. Section 4.1). In Section G.2, we detail the instance generation for our real-world experiments (cf. Section 4.2).

G.1. Computational performance experiments

To compare our skeleton algorithm against the algorithm of Pansart et al. (2018) for a picker routing problem with a multiblock warehouse limited to a single picklist, a single drop-off point, a static batching, and no cartless subtours, we use the benchmark from Theys et al. (2010). This benchmark comprises 1,080 instances which vary with respect to the number of aisles (5, 15, 60), the number of cross-aisles (3, 6, 11), the number of picks (15, 60, 240), a volume-based (V) or random (R) storage policy, and central as well as decentralized drop-off point positions.

To generate picklists for our additional instances, we randomly choose pick positions to create a sequence of picklists and consider 20–60 picks per picklist, which correspond to a typical bin capacity for online retailers selling small to mid-size consumer goods. We note that the derived picklists do not allow for meaningful managerial anaylsis as an upstream batching procedure would derive more efficient picklists that are less spatially dispersed within a picking zone. However, a randomly distributed picklist poses a bigger challenge to our algorithm as the number of mandatory edges is higher. Accordingly, we use this picklist generation scheme only to derive instances for conservative, pure computational studies.

G.2. Real-world setting & optimal picking policies

Through this experiment, we aim to derive insights into the optimal design of picking policies and the respective picking zone layouts. To this end, we perform a full factorial design with respect to the picking policy characteristics (Table 1) and the picking zone layouts from Section 4.1.2. To complete the experiment, we extend our analysis as follows:

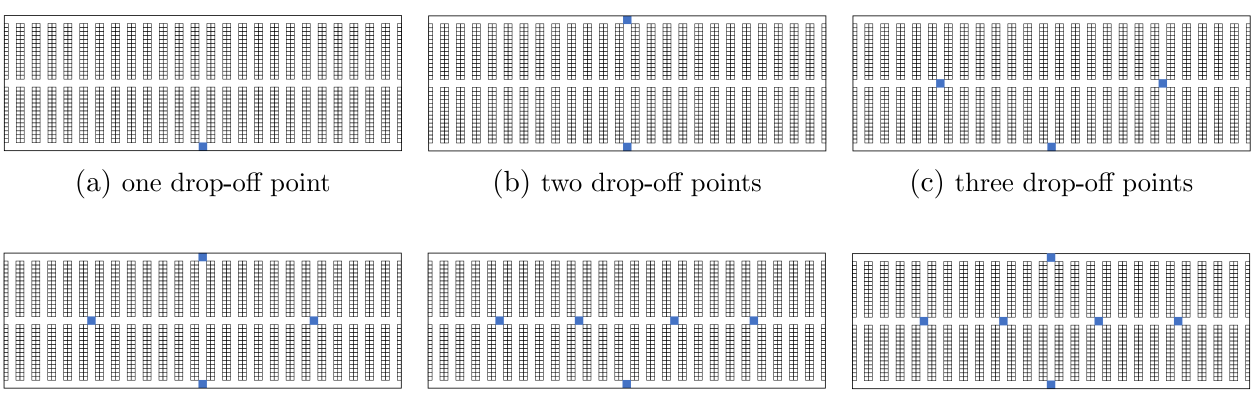
i) Taking the real-world layouts as an envelope to our study, we design picking zone layouts with one to six drop-off points and three and five cross-aisles. By enlarging the number of settings for additional drop-off point configurations beyond the configurations observed in practice, our study allows to derive more general insights into the benefit of picking zone layouts. Figure 21 shows an example of the drop-off point configurations for a layout with three cross-aisles.

ii) For all resulting layouts, we evaluate the impact of a) cartless subtours and b) dynamic batching. Note that we implicitly consider multiple drop-off points by each picking zone layout.

iii) The number of available bins, and therefore the number of parallel processed picklists in a dynamic batching policy varies between two (Amazon, Zalando) and four (Hermes). Hence, we add an additional parameter scan over the number of available bins $\kappa \in \{2, 3, 4\}$, but we do not extend it to larger numbers of bins as the cart size is limited in practice.

For cartless subtours, we assume a picker walking speed of 5 km/h, whereas this speed goes down to 4 km/h if she is accompanied by her cart. We do not consider further variations in the picker walking speed as these values reflect physical properties which are hardly alterable.

Besides these structural characteristics, the dispersion of a picklist's pick positions, which results from the efficiency of an upstream batching procedure, might significantly impact the efficiency of specific picking policies. To take this effect into account, we generate picklist sequences with different levels of dispersion. Our instance generation scheme bases on two ideas to mimic an upstream batching procedure. First, we take into account that in a scattered multi-depot warehouse an efficient batching aims to create a picklist sequence such that the pick positions of subsequent picklists have a maximum overlap (Weidinger et al. 2019). Second, we consider that an efficient batching aims to minimize the distance between the pick positions within each picklist, i.e., a picklist's level of dispersion. Note that the picklist dispersion is only one criterion for the batch building process. For an elaboration on further criteria, e.g., the orders' urgency and a fair division of workload among zones, see Boysen et al. (2019). If more weight is attributed to these further criteria, dispersed picklists need not be an indicator for a flawed batch building process. Specifically, our instance generation scheme accounts for these characteristics as follows.



(a) one drop-off point (b) two drop-off points (c) three drop-off points

(d) four drop-off points (e) five drop-off points (f) six drop-off points

Figure 21 Example of the different drop-off point configurations for a layout with three cross-aisles.

We first account for creating an efficient list sequence that is still subject to some random order behavior as follows.

1. To create a picklist sequence with n subsequent picklists, we randomly draw $3n$ pick positions within the picking zone layout, each representing the center of a certain picklist.
2. We then select a cluster of n of these positions by randomly selecting one position and choosing $n-1$ additional positions such that the total distance between the positions in the cluster is minimal.

We then account for a picklist's level of dispersion by limiting the pick positions of a picklist to a certain area around its center in the picking zone. By varying this area we can then steer the level of dispersion for each picklist.

3. To determine the pick positions of each picklist, we randomly draw pick positions within a limited area of the picking zone, which we define based on the prior selected center of each picklist. Specifically, we limit the area from which pick positions can be drawn by the number of aisles $\beta \in \mathbb{N}$ to the left and right of the center's aisle and to the share of pick positions within an aisle $\gamma \in [0,1]$ above and below the center's position, cf. Figure 22. Accordingly, a picklist that has been created with a small β and γ reflects an efficient upstream batching, while a picklist that has been created with a large β and γ is the result of a less efficient batching where picking positions are spread over a larger area.
4. For each picklist, we randomly draw 35 positions within the respective area. By so doing, we account for a typical bin capacity for online retailers selling small to mid-size consumer goods.

To account for diferent efficient upstream batching procedures, we create instances with picklist neighborhoods in the range of $\beta \in \{2,4,6,8,12,16\}$ and $\gamma \in \{0.25,0.5,0.75,1.0\}$. We consider a full factorial design over β and γ and create 50 instances for each setting to avoid statistical bias.

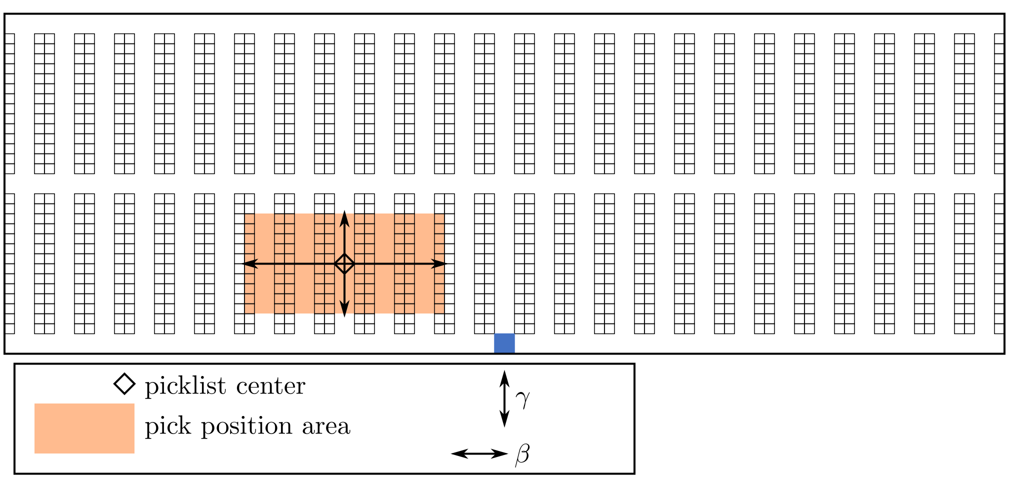


Figure 22 Example of a picklist neighborhood.

Appendix H: Preliminary analyses of cart capacity impact

In the following, we exemplary discuss the benefit of increasing the cart capacity for the most simple picking policy with static batching, a single drop-off point, and no cartless subtours (SB/SD/NS), which allows for a clean analysis that is not influenced by enhanced picking policies. To this end, we generate instances as described in Section G.2 and vary the number of consecutive picklists, considering instances with 5, 20, 35, and 50 consecutive picklists. We create 50 instances for each consecutive picklist setting, picklist dispersion for three- and five-cross-aisle layouts.

To analyze the impact of an increased cart capacity, we calculate the total picking time reduction (TPTR) when increasing the cart capacity from $\kappa = 2$ to $\kappa = 3$ (Δ_{23}) or from $\kappa = 2$ to $\kappa = 4$ (Δ_{24}) related to the total picking time with $\kappa = 2$. Additionally, we denote the improvement that can be realized by further increasing the cart capacity from $\kappa = 3$ to $\kappa = 4$ as $\Delta_{34} = \Delta_{24} - \Delta_{23}$. Analogously to Section 4.2.2, we report results based on average relative reductions which have been calculated by first comparing the results on a per instance basis and then computing an average value.

Figure 23 shows the average TPTR reduction when increasing the cart capacity from $\kappa = 2$ to $\kappa = 3$ or from $\kappa = 3$ to $\kappa = 4$ when using an SB/SD/NS policy on a three-cross-aisle layout with a single drop-off point for five consecutively processed picklists. As can be seen average TPTRs of up to $\Delta_{24} = 30\%$ can be realized by switching the cart capacity to $\kappa = 4$. However, we observe an imbalance between Δ_{23} and Δ_{34} as only up to 8% improvement can be realized by increasing the cart capacity from $\kappa = 3$ to $\kappa = 4$. A detailed analysis reveals that this imbalance between Δ_{23} and Δ_{34} stems from a termination effect, where on the final picking tour, the cart cannot be loaded with bins to capacity due to the limited number of consecutive picklists.

To exclude this termination effect, we analyze the resulting TPTRs for instances with a larger number of consecutive picklists in Figure 24. Here, we observe rather stable results for settings with 20, 35, and 50 picklists, which shows that the chosen number of picklists is sufficient to remedy a termination effect. Here, we observe average TPTR improvements of up to $\Delta_{24} = 44\%$ with up to $\Delta_{34} = 16\%$ resulting from increasing the cart capacity from $\kappa = 3$ to $\kappa = 4$. While these numbers could be interpreted as a possible improvement potential of increasing cart capacity, two comments on these findings are in order.

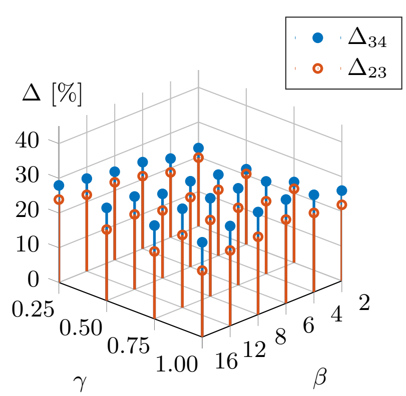
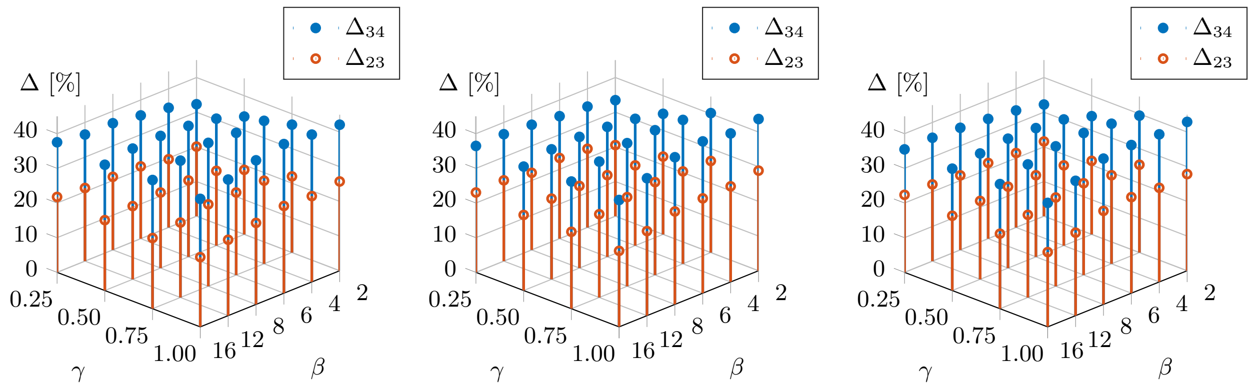


Figure 23 Average TPTR for SB/SD/NS on a three-cross-aisle layout with one drop-off point and five consecutive picklists.



(a) Average TPTR for 20 consecutive picklists.

(b) Average TPTR for 35 consecutive picklists.

(c) Average TPTR for 50 consecutive picklists.

Figure 24 Average TPTR for SB/SD/NS on a three-cross-aisle layout with one drop-off point dependent on the number of consecutive picklists.

First, our results reveal that increasing cart capacity promises considerable improvements because the picking density per tour improves with a larger cart capacity. However, in the narrow aisles of a typical e-commerce warehouse, automated picking carts are not applicable, and handling manual carts for multiple large batch picking bins burdens the pickers with excessive physical stress. For example, at the Hermes warehouse (see Section 2.2), where they apply a cart capacity of $\kappa = 4$, they mainly handle light-weighed apparel. According to our experience, for heavier and bulkier goods such a cart capacity risks the physical health of the human workforce over the long run and considerably slows down the travel speed in the narrow aisles. Thus, the huge improvements of larger carts promised by our experiments must be put into perspective with these negative effects on the human workforce, and more research on this trade-off is necessary.

Second, to remain reactive on unforeseen change, it is our experience, that most online retailers apply fairly short planning horizons, e.g., not exceeding planning intervals of fifteen minutes, which is reflected in our original experimental design with five consecutive picklists. Accordingly, a straighforward extension of the instances' to larger numbers of consecutive picklists allows to remedy the initial termination effect but remains a simplified approximation of analyzing the improvement potential of increased cart capacities. In practice, retailers may apply more enhanced rolling horizon planning schemes. Analyzing the impact of increased cart capacities in such realistic schemes in detail goes beyond the scope of this paper and would require a significantly different experimental design.

Appendix I: Detailed numerical results

This appendix shows all numerical results that have been synthesized in Section 4.2.2. We report results based on average relative improvements, which have been calculated by first comparing the results on a per-instance basis and then computing an average value.

We describe each picking policy with three identifiers, where the first (SB,DB) indicates whether batching is static (SB) or dynamic (DB), the second (SD,MD) indicates whether a single (SD) or multiple (MD) drop-off points are used, and the third (NS,CS) indicates whether cartless subtours are allowed (CS) or not (NS).

To quantify the improvement potential of each picking zone layout and policy, we show the average relative improvement in the total picking time for each setting and κ related to the average total picking time realized with a single depot and the most elementary picking policy (SB/SD/NS). Note that if a policy does not allow for multiple drop-off points, the comparison is skipped.

I.1. Results for cart capacity $\kappa = 2$

Table 6 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 2$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-1.1	-	-	-	-	-
SB/SD/CS	-3.3	-	-	-	-	-	-4.0	-	-	-	-	-	-1.8	-	-	-	-	-
SB/MD/NS	0.0	-8.5	-18.4	-18.9	-20.1	-20.4	0.0	-11.2	-21.8	-22.7	-23.6	-24.3	-1.1	-4.3	-5.4	-6.0	-5.4	-6.1
SB/MD/CS	-3.3	-11.7	-22.3	-22.7	-24.1	-24.4	-4.0	-14.7	-25.9	-26.7	-27.6	-28.3	-1.8	-4.7	-5.8	-6.3	-5.8	-6.3
DB/SD/NS	-3.4	-	-	-	-	-	-3.1	-	-	-	-	-	-0.8	-	-	-	-	-
DB/SD/CS	-6.4	-	-	-	-	-	-6.9	-	-	-	-	-	-1.6	-	-	-	-	-
DB/MD/NS	-3.4	-12.2	-22.2	-22.9	-23.8	-24.3	-3.1	-14.6	-24.9	-25.8	-26.6	-27.4	-0.8	-4.0	-4.6	-4.9	-4.7	-5.2
DB/MD/CS	-6.4	-15.2	-25.5	-26.1	-27.2	-27.6	-6.9	-17.8	-28.7	-29.5	-30.4	-31.1	-1.6	-4.4	-5.3	-5.6	-5.4	-5.8

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 7 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 4$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-3.1	-	-	-	-	-
SB/SD/CS	-5.5	-	-	-	-	-	-5.2	-	-	-	-	-	-2.8	-	-	-	-	-
SB/MD/NS	0.0	-5.1	-11.3	-12.3	-12.8	-13.6	0.0	-5.7	-14.4	-15.1	-15.8	-16.4	-3.1	-3.7	-6.4	-6.1	-6.3	-6.1
SB/MD/CS	-5.5	-10.2	-17.2	-17.9	-18.8	-19.3	-5.2	-9.8	-19.4	-19.8	-20.9	-21.2	-2.8	-2.7	-5.7	-5.4	-5.7	-5.4
DB/SD/NS	-4.5	-	-	-	-	-	-4.4	-	-	-	-	-	-3.0	-	-	-	-	-
DB/SD/CS	-9.2	-	-	-	-	-	-9.3	-	-	-	-	-	-3.2	-	-	-	-	-
DB/MD/NS	-4.5	-9.7	-16.0	-17.1	-17.4	-18.3	-4.4	-10.0	-18.6	-19.2	-20.0	-20.5	-3.0	-3.4	-6.1	-5.5	-6.1	-5.6
DB/MD/CS	-9.2	-14.0	-21.1	-22.0	-22.5	-23.2	-9.3	-13.7	-23.4	-23.6	-24.8	-24.9	-3.2	-2.8	-5.9	-5.2	-5.9	-5.3

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 8 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 6$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-2.8	-	-	-	-	-
SB/SD/CS	-4.7	-	-	-	-	-	-5.5	-	-	-	-	-	-3.6	-	-	-	-	-
SB/MD/NS	0.0	-3.9	-7.5	-8.3	-8.6	-9.2	0.0	-4.5	-9.4	-10.2	-10.6	-11.2	-2.8	-3.5	-4.9	-4.9	-5.1	-5.0
SB/MD/CS	-4.7	-7.2	-11.9	-12.5	-13.0	-13.5	-5.5	-8.7	-14.8	-15.2	-16.1	-16.3	-3.6	-4.5	-6.1	-5.9	-6.2	-6.0
DB/SD/NS	-4.1	-	-	-	-	-	-3.3	-	-	-	-	-	-2.0	-	-	-	-	-
DB/SD/CS	-8.4	-	-	-	-	-	-8.6	-	-	-	-	-	-3.0	-	-	-	-	-
DB/MD/NS	-4.1	-8.1	-11.6	-12.7	-12.7	-13.6	-3.3	-8.3	-13.5	-14.4	-14.7	-15.4	-2.0	-3.0	-4.9	-4.7	-5.0	-4.8
DB/MD/CS	-8.4	-11.1	-15.6	-16.4	-16.8	-17.3	-8.6	-12.2	-18.5	-19.0	-19.7	-20.1	-3.0	-4.0	-6.1	-5.8	-6.3	-6.0

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 9 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 8$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	1.4	-	-	-	-	-
SB/SD/CS	-4.5	-	-	-	-	-	-4.6	-	-	-	-	-	1.2	-	-	-	-	-
SB/MD/NS	0.0	-6.0	-6.4	-8.5	-7.4	-9.0	0.0	-3.5	-5.5	-6.6	-6.3	-7.3	1.4	4.1	2.4	3.4	2.5	3.3
SB/MD/CS	-4.5	-7.5	-9.9	-10.8	-10.7	-11.5	-4.6	-6.6	-9.7	-10.2	-10.5	-10.9	1.2	2.5	1.6	2.1	1.6	2.1
DB/SD/NS	-4.5	-	-	-	-	-	-4.9	-4.9	-	-	-	-	-	-	-	-	-	-
DB/SD/CS	-8.7	-	-	-	-	-	-9.4	-9.4	-	-	-	-	-	-	-	-	-	-
DB/MD/NS	-4.5	-11.1	-11.3	-13.6	-12.3	-14.2	-4.9	-8.5	-10.6	-11.8	-11.5	-12.4	0.9	4.4	2.2	3.5	2.4	3.5
DB/MD/CS	-8.7	-12.3	-14.4	-15.6	-15.2	-16.3	-9.4	-11.3	-14.4	-15.0	-15.1	-15.6	0.7	2.7	1.4	2.1	1.5	2.2

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 10 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 12$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-6.1	-	-	-	-	-
SB/SD/CS	-5.0	-	-	-	-	-	-5.7	-	-	-	-	-	-6.8	-	-	-	-	-
SB/MD/NS	0.0	-6.1	-13.7	-14.0	-14.8	-15.0	0.0	-8.4	-16.6	-17.2	-17.8	-18.2	-6.1	-8.8	-9.5	-9.8	-9.5	-9.8
SB/MD/CS	-5.0	-11.0	-19.2	-19.4	-20.3	-20.5	-5.7	-14.0	-22.5	-23.0	-23.7	-24.1	-6.8	-9.5	-10.1	-10.5	-10.2	-10.5
DB/SD/NS	-4.6	-	-	-	-	-	-4.4	-	-	-	-	-	-6.0	-	-	-	-	-
DB/SD/CS	-8.8	-	-	-	-	-	-9.5	-	-	-	-	-	-6.8	-	-	-	-	-
DB/MD/NS	-4.6	-10.7	-18.6	-19.0	-19.7	-20.0	-4.4	-13.3	-21.2	-21.9	-22.3	-22.8	-6.0	-9.1	-9.2	-9.6	-9.2	-9.6
DB/MD/CS	-8.8	-15.0	-23.2	-23.6	-24.3	-24.6	-9.5	-18.1	-26.3	-26.9	-27.4	-27.8	-6.8	-9.8	-10.0	-10.3	-10.0	-10.3

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 11 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 16$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-6.5	-	-	-	-	-
SB/SD/CS	-6.8	-	-	-	-	-	-5.9	-	-	-	-	-	-5.7	-	-	-	-	-
SB/MD/NS	0.0	-2.9	-8.0	-8.4	-8.9	-9.2	0.0	-3.5	-9.8	-9.9	-10.6	-10.7	-6.5	-7.1	-8.3	-8.1	-8.3	-8.1
SB/MD/CS	-6.8	-9.8	-15.4	-15.5	-16.3	-16.4	-5.9	-9.0	-15.8	-16.0	-16.7	-16.8	-5.7	-5.8	-7.2	-7.1	-7.1	-7.1
DB/SD/NS	-4.5	-	-	-	-	-	-4.2	-	-	-	-	-	-6.2	-	-	-	-	-
DB/SD/CS	-10.5	-	-	-	-	-	-9.7	-	-	-	-	-	-5.7	-	-	-	-	-
DB/MD/NS	-4.5	-7.4	-12.9	-13.3	-13.7	-14.0	-4.2	-7.6	-13.9	-14.1	-14.7	-14.8	-6.2	-6.7	-7.6	-7.3	-7.6	-7.4
DB/MD/CS	-10.5	-13.2	-19.1	-19.4	-20.0	-20.2	-9.7	-12.8	-19.4	-19.5	-20.3	-20.3	-5.7	-6.1	-7.0	-6.9	-6.9	-6.8

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 12 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 2$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-8.5	-	-	-	-	-
SB/SD/CS	-6.2	-	-	-	-	-	-6.1	-	-	-	-	-	-8.4	-	-	-	-	-
SB/MD/NS	0.0	-1.3	-4.5	-4.7	-5.1	-5.3	0.0	-2.3	-6.3	-6.5	-6.9	-7.0	-8.5	-9.4	-10.2	-10.2	-10.3	-10.2
SB/MD/CS	-6.2	-7.1	-11.0	-11.1	-11.6	-11.7	-6.1	-8.2	-12.6	-12.8	-13.2	-13.3	-8.4	-9.5	-10.2	-10.2	-10.2	-10.2
DB/SD/NS	-5.9	-	-	-	-	-	-5.3	-	-	-	-	-	-7.8	-	-	-	-	-
DB/SD/CS	-11.1	-	-	-	-	-	-10.5	-	-	-	-	-	-7.8	-	-	-	-	-
DB/MD/NS	-5.9	-7.2	-10.7	-10.9	-11.2	-11.5	-5.3	-7.2	-11.6	-11.8	-12.2	-12.3	-7.8	-8.5	-9.3	-9.3	-9.4	-9.3
DB/MD/CS	-11.1	-12.2	-16.1	-16.3	-16.8	-16.9	-10.5	-12.4	-17.2	-17.3	-17.8	-17.8	-7.8	-8.6	-9.6	-9.5	-9.5	-9.4

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 13 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 4$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-2.0	-	-	-	-	-
SB/SD/CS	-4.3	-	-	-	-	-	-5.1	-	-	-	-	-	-2.8	-	-	-	-	-
SB/MD/NS	0.0	-1.4	-3.8	-4.2	-4.4	-4.6	0.0	-1.2	-3.5	-3.8	-4.0	-4.2	-2.0	-1.7	-1.6	-1.6	-1.5	-1.5
SB/MD/CS	-4.3	-5.2	-7.9	-8.1	-8.4	-8.6	-5.1	-6.5	-8.9	-9.2	-9.3	-9.5	-2.8	-3.3	-3.1	-3.2	-3.0	-3.0
DB/SD/NS	-4.2	-	-	-	-	-	-3.8	-	-	-	-	-	-1.5	-	-	-	-	-
DB/SD/CS	-7.9	-	-	-	-	-	-8.5	-	-	-	-	-	-2.6	-	-	-	-	-
DB/MD/NS	-4.2	-5.8	-8.3	-8.7	-8.9	-9.2	-3.8	-5.0	-7.5	-7.8	-7.9	-8.1	-1.5	-1.1	-1.0	-0.9	-0.9	-0.8
DB/MD/CS	-7.9	-8.9	-11.9	-12.1	-12.3	-12.6	-8.5	-9.8	-12.4	-12.7	-12.8	-13.0	-2.6	-3.0	-2.6	-2.6	-2.4	-2.4

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 14 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 6$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.8	-	-	-	-	-
SB/SD/CS	-7.2	-	-	-	-	-	-6.4	-	-	-	-	-	-10.9	-	-	-	-	-
SB/MD/NS	0.0	-4.6	-10.5	-10.6	-11.4	-11.5	0.0	-5.8	-13.4	-13.7	-14.2	-14.4	-11.8	-13.0	-14.8	-14.9	-14.7	-14.8
SB/MD/CS	-7.2	-11.9	-18.2	-18.3	-19.1	-19.1	-6.4	-12.0	-19.9	-20.1	-20.7	-20.8	-10.9	-12.0	-13.6	-13.8	-13.6	-13.7
DB/SD/NS	-4.5	-	-	-	-	-	-2.9	-	-	-	-	-	-10.3	-	-	-	-	-
DB/SD/CS	-10.7	-	-	-	-	-	-8.7	-	-	-	-	-	-9.8	-	-	-	-	-
DB/MD/NS	-4.5	-8.8	-15.3	-15.4	-16.2	-16.2	-2.9	-8.7	-16.7	-16.9	-17.6	-17.7	-10.3	-11.7	-13.2	-13.3	-13.3	-13.3
DB/MD/CS	-10.7	-15.0	-22.0	-22.1	-22.8	-22.9	-8.7	-14.2	-22.7	-22.8	-23.4	-23.5	-9.8	-11.0	-12.5	-12.5	-12.5	-12.5

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 15 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 8$, $\gamma = 0.5$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-8.8	-	-	-	-	-
SB/SD/CS	-7.9	-	-	-	-	-	-6.5	-	-	-	-	-	-7.5	-	-	-	-	-
SB/MD/NS	0.0	-2.9	-6.5	-6.8	-7.3	-7.4	0.0	-3.6	-8.4	-8.5	-9.1	-9.2	-8.8	-9.6	-10.6	-10.5	-10.6	-10.5
SB/MD/CS	-7.9	-11.0	-15.1	-15.3	-16.0	-16.0	-6.5	-10.2	-15.1	-15.2	-15.9	-15.9	-7.5	-8.1	-8.9	-8.9	-8.8	-8.8
DB/SD/NS	-4.1	-	-	-	-	-	-3.8	-	-	-	-	-	-8.5	-	-	-	-	-
DB/SD/CS	-11.1	-	-	-	-	-	-9.8	-	-	-	-	-	-7.5	-	-	-	-	-
DB/MD/NS	-4.1	-7.3	-10.9	-11.2	-11.7	-11.9	-3.8	-7.6	-12.5	-12.7	-13.2	-13.3	-8.5	-9.1	-10.4	-10.3	-10.3	-10.2
DB/MD/CS	-11.1	-14.3	-18.5	-18.6	-19.3	-19.4	-9.8	-13.5	-18.8	-18.9	-19.4	-19.4	-7.5	-8.1	-9.2	-9.2	-8.9	-8.9

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 16 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 12$, $\gamma = 0.5$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-12.3	-	-	-	-	-
SB/SD/CS	-7.2	-	-	-	-	-	-6.8	-	-	-	-	-	-11.9	-	-	-	-	-
SB/MD/NS	0.0	-0.7	-2.9	-3.0	-3.5	-3.5	0.0	-2.2	-4.9	-5.1	-5.3	-5.5	-12.3	-13.6	-14.1	-14.2	-14.0	-14.1
SB/MD/CS	-7.2	-8.4	-10.6	-10.8	-11.2	-11.3	-6.8	-9.3	-12.1	-12.3	-12.5	-12.6	-11.9	-13.2	-13.8	-13.8	-13.6	-13.6
DB/SD/NS	-3.8	-	-	-	-	-	-2.8	-	-	-	-	-	-11.4	-	-	-	-	-
DB/SD/CS	-10.1	-	-	-	-	-	-9.3	-	-	-	-	-	-11.5	-	-	-	-	-
DB/MD/NS	-3.8	-4.8	-7.1	-7.2	-7.6	-7.7	-2.8	-5.4	-7.9	-8.3	-8.4	-8.6	-11.4	-12.8	-13.1	-13.3	-13.0	-13.2
DB/MD/CS	-10.1	-11.4	-13.6	-13.9	-14.2	-14.4	-9.3	-11.9	-14.8	-15.1	-15.3	-15.5	-11.5	-12.9	-13.5	-13.6	-13.4	-13.3

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 17 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 16$, $\gamma = 0.5$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-6.4	-	-	-	-	-
SB/SD/CS	-5.6	-	-	-	-	-	-6.4	-	-	-	-	-	-7.2	-	-	-	-	-
SB/MD/NS	0.0	-0.8	-3.0	-3.1	-3.3	-3.4	0.0	-1.3	-3.1	-3.3	-3.3	-3.5	-6.4	-6.9	-6.5	-6.6	-6.4	-6.5
SB/MD/CS	-5.6	-6.5	-8.3	-8.4	-8.6	-8.8	-6.4	-7.8	-9.7	-9.9	-9.9	-10.1	-7.2	-7.7	-7.8	-7.9	-7.7	-7.8
DB/SD/NS	-4.3	-	-	-	-	-	-3.9	-	-	-	-	-	-6.0	-	-	-	-	-
DB/SD/CS	-9.4	-	-	-	-	-	-9.9	-	-	-	-	-	-6.9	-	-	-	-	-
DB/MD/NS	-4.3	-5.1	-7.5	-7.7	-7.9	-8.0	-3.9	-5.2	-7.1	-7.4	-7.5	-7.7	-6.0	-6.5	-6.0	-6.1	-5.9	-6.0
DB/MD/CS	-9.4	-10.4	-12.4	-12.5	-12.8	-12.9	-9.9	-11.1	-13.2	-13.4	-13.4	-13.5	-6.9	-7.2	-7.2	-7.3	-7.0	-7.1

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 18 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 2$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.8	-	-	-	-	-
SB/SD/CS	-8.7	-	-	-	-	-	-7.3	-	-	-	-	-	-10.3	-	-	-	-	-
SB/MD/NS	0.0	-3.8	-8.9	-9.0	-9.7	-9.7	0.0	-5.3	-12.0	-12.2	-12.8	-12.9	-11.8	-13.3	-14.8	-15.0	-14.8	-15.0
SB/MD/CS	-8.7	-12.5	-18.2	-18.2	-18.9	-18.9	-7.3	-12.5	-19.5	-19.7	-20.3	-20.4	-10.3	-11.9	-13.2	-13.4	-13.2	-13.3
DB/SD/NS	-2.5	-	-	-	-	-	-2.5	-	-	-	-	-	-11.7	-	-	-	-	-
DB/SD/CS	-10.8	-	-	-	-	-	-9.3	-	-	-	-	-	-10.2	-	-	-	-	-
DB/MD/NS	-2.5	-6.3	-11.5	-11.6	-12.3	-12.3	-2.5	-7.9	-14.5	-14.8	-15.4	-15.6	-11.7	-13.3	-14.8	-15.1	-14.9	-15.1
DB/MD/CS	-10.8	-14.5	-20.3	-20.4	-21.0	-21.0	-9.3	-14.5	-21.6	-21.9	-22.4	-22.6	-10.2	-11.8	-13.1	-13.4	-13.2	-13.4

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 19 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 4$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-12.0	-	-	-	-	-
SB/SD/CS	-9.3	-	-	-	-	-	-7.5	-	-	-	-	-	-10.4	-	-	-	-	-
SB/MD/NS	0.0	-2.3	-5.0	-5.2	-5.6	-5.7	0.0	-3.5	-7.2	-7.2	-7.7	-7.7	-12.0	-13.2	-14.1	-13.9	-14.0	-13.9
SB/MD/CS	-9.3	-11.8	-15.0	-15.0	-15.6	-15.6	-7.5	-11.0	-14.6	-14.7	-15.1	-15.1	-10.4	-11.4	-11.8	-11.8	-11.7	-11.7
DB/SD/NS	-4.0	-	-	-	-	-	-3.0	-	-	-	-	-	-11.2	-	-	-	-	-
DB/SD/CS	-12.4	-	-	-	-	-	-10.1	-	-	-	-	-	-9.8	-	-	-	-	-
DB/MD/NS	-4.0	-6.3	-9.1	-9.2	-9.6	-9.6	-3.0	-6.6	-10.4	-10.5	-11.0	-11.1	-11.2	-12.3	-13.3	-13.2	-13.3	-13.3
DB/MD/CS	-12.4	-15.0	-18.0	-18.1	-18.6	-18.7	-10.1	-13.7	-17.4	-17.5	-18.0	-18.0	-9.8	-10.9	-11.5	-11.5	-11.4	-11.4

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 20 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 6$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.6	-	-	-	-	-
SB/SD/CS	-8.6	-	-	-	-	-	-7.8	-	-	-	-	-	-15.0	-	-	-	-	-
SB/MD/NS	0.0	-0.6	-2.0	-2.2	-2.4	-2.5	0.0	-2.1	-3.4	-3.6	-3.8	-3.9	-15.6	-16.9	-16.9	-16.8	-16.9	-16.8
SB/MD/CS	-8.6	-9.6	-10.9	-11.1	-11.3	-11.4	-7.8	-10.1	-11.6	-11.8	-11.9	-12.0	-15.0	-16.1	-16.2	-16.2	-16.2	-16.2
DB/SD/NS	-3.9	-	-	-	-	-	-3.6	-	-	-	-	-	-15.3	-	-	-	-	-
DB/SD/CS	-11.7	-	-	-	-	-	-10.7	-	-	-	-	-	-14.7	-	-	-	-	-
DB/MD/NS	-3.9	-4.6	-6.1	-6.2	-6.5	-6.6	-3.6	-5.7	-7.0	-7.3	-7.5	-7.7	-15.3	-16.5	-16.4	-16.6	-16.5	-16.6
DB/MD/CS	-11.7	-12.8	-14.1	-14.3	-14.5	-14.7	-10.7	-13.1	-14.6	-14.9	-14.9	-15.1	-14.7	-15.9	-16.1	-16.2	-16.0	-16.0

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 21 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 8$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.0	-	-	-	-	-
SB/SD/CS	-6.9	-	-	-	-	-	-7.1	-	-	-	-	-	-9.3	-	-	-	-	-
SB/MD/NS	0.0	-0.6	-1.7	-1.8	-1.9	-2.0	0.0	-0.9	-1.5	-1.7	-1.7	-1.9	-9.0	-9.4	-8.9	-8.9	-8.9	-8.9
SB/MD/CS	-6.9	-7.6	-8.2	-8.4	-8.5	-8.6	-7.1	-8.3	-8.7	-9.1	-9.0	-9.2	-9.3	-9.7	-9.5	-9.7	-9.6	-9.7
DB/SD/NS	-4.2	-	-	-	-	-	-4.2	-	-	-	-	-	-9.0	-	-	-	-	-
DB/SD/CS	-10.7	-	-	-	-	-	-10.9	-	-	-	-	-	-9.2	-	-	-	-	-
DB/MD/NS	-4.2	-4.9	-6.1	-6.3	-6.4	-6.6	-4.2	-5.3	-5.8	-6.2	-6.2	-6.5	-9.0	-9.4	-8.7	-8.9	-8.8	-8.9
DB/MD/CS	-10.7	-11.5	-12.1	-12.4	-12.5	-12.7	-10.9	-12.0	-12.5	-12.9	-12.8	-13.1	-9.2	-9.6	-9.4	-9.6	-9.3	-9.5

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 22 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 12$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-14.9	-	-	-	-	-
SB/SD/CS	-10.3	-	-	-	-	-	-8.2	-	-	-	-	-	-12.8	-	-	-	-	-
SB/MD/NS	0.0	-3.3	-7.6	-7.7	-8.4	-8.5	0.0	-4.5	-10.2	-10.3	-10.9	-11.0	-14.9	-16.0	-17.3	-17.4	-17.2	-17.3
SB/MD/CS	-10.3	-13.6	-18.5	-18.5	-19.2	-19.2	-8.2	-12.8	-18.5	-18.6	-19.2	-19.3	-12.8	-14.0	-14.8	-14.9	-14.8	-14.8
DB/SD/NS	-3.2	-	-	-	-	-	-3.3	-	-	-	-	-	-15.0	-	-	-	-	-
DB/SD/CS	-12.9	-	-	-	-	-	-11.1	-	-	-	-	-	-13.0	-	-	-	-	-
DB/MD/NS	-3.2	-6.6	-10.9	-10.9	-11.6	-11.6	-3.3	-7.9	-13.5	-13.6	-14.2	-14.3	-15.0	-16.1	-17.4	-17.4	-17.4	-17.5
DB/MD/CS	-12.9	-16.1	-21.1	-21.1	-21.8	-21.8	-11.1	-15.6	-21.3	-21.4	-22.1	-22.2	-13.0	-14.3	-14.9	-15.0	-15.0	-15.1

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 23 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 16$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.7	-	-	-	-	-
SB/SD/CS	-10.9	-	-	-	-	-	-8.9	-	-	-	-	-	-13.9	-	-	-	-	-
SB/MD/NS	0.0	-1.2	-3.6	-3.6	-4.1	-4.2	0.0	-2.1	-5.1	-5.1	-5.6	-5.6	-15.7	-16.5	-17.0	-17.0	-17.0	-17.0
SB/MD/CS	-10.9	-12.4	-15.0	-15.0	-15.5	-15.6	-8.9	-11.1	-13.9	-14.0	-14.4	-14.4	-13.9	-14.5	-14.7	-14.7	-14.6	-14.6
DB/SD/NS	-3.3	-	-	-	-	-	-2.8	-	-	-	-	-	-15.3	-	-	-	-	-
DB/SD/CS	-13.5	-	-	-	-	-	-11.2	-	-	-	-	-	-13.5	-	-	-	-	-
DB/MD/NS	-3.3	-4.6	-7.1	-7.2	-7.7	-7.7	-2.8	-4.9	-8.0	-8.1	-8.7	-8.7	-15.3	-16.0	-16.5	-16.4	-16.6	-16.5
DB/MD/CS	-13.5	-15.0	-17.7	-17.8	-18.3	-18.4	-11.2	-13.5	-16.4	-16.5	-16.8	-16.8	-13.5	-14.3	-14.4	-14.4	-14.1	-14.1

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 24 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 2$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-20.8	-	-	-	-	-
SB/SD/CS	-10.6	-	-	-	-	-	-8.7	-	-	-	-	-	-19.2	-	-	-	-	-
SB/MD/NS	0.0	-0.4	-1.4	-1.5	-1.7	-1.8	0.0	-1.2	-2.2	-2.4	-2.5	-2.6	-20.8	-21.5	-21.5	-21.5	-21.4	-21.4
SB/MD/CS	-10.6	-11.3	-11.9	-12.1	-12.2	-12.3	-8.7	-10.0	-11.2	-11.4	-11.5	-11.6	-19.2	-19.7	-20.2	-20.1	-20.2	-20.1
DB/SD/NS	-3.7	-	-	-	-	-	-3.1	-	-	-	-	-	-20.3	-	-	-	-	-
DB/SD/CS	-13.6	-	-	-	-	-	-11.4	-	-	-	-	-	-18.8	-	-	-	-	-
DB/MD/NS	-3.7	-4.1	-5.2	-5.3	-5.6	-5.7	-3.1	-4.5	-5.5	-5.6	-5.7	-5.8	-20.3	-21.1	-21.0	-21.0	-20.9	-20.9
DB/MD/CS	-13.6	-14.2	-15.0	-15.1	-15.3	-15.4	-11.4	-12.7	-13.9	-14.0	-14.3	-14.3	-18.8	-19.4	-19.8	-19.7	-19.8	-19.7

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 25 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 4$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-13.6	-	-	-	-	-
SB/SD/CS	-8.6	-	-	-	-	-	-8.0	-	-	-	-	-	-13.0	-	-	-	-	-
SB/MD/NS	0.0	-0.5	-1.0	-1.1	-1.2	-1.3	0.0	-1.0	-1.0	-1.3	-1.2	-1.4	-13.6	-14.1	-13.7	-13.8	-13.7	-13.8
SB/MD/CS	-8.6	-9.4	-9.4	-9.6	-9.5	-9.7	-8.0	-9.1	-9.1	-9.4	-9.3	-9.5	-13.0	-13.4	-13.3	-13.4	-13.4	-13.5
DB/SD/NS	-3.8	-	-	-	-	-	-3.5	-	-	-	-	-	-13.4	-	-	-	-	-
DB/SD/CS	-12.0	-	-	-	-	-	-10.7	-	-	-	-	-	-12.4	-	-	-	-	-
DB/MD/NS	-3.8	-4.4	-5.0	-5.2	-5.2	-5.3	-3.5	-4.5	-4.5	-4.9	-4.7	-5.0	-13.4	-13.8	-13.2	-13.4	-13.2	-13.4
DB/MD/CS	-12.0	-12.7	-12.9	-13.1	-13.0	-13.2	-10.7	-11.9	-11.9	-12.3	-12.1	-12.4	-12.4	-12.8	-12.7	-12.9	-12.8	-12.9

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 26 Average policy and layout improvement potential with $\kappa = 2$, $\beta = 6$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.9	-	-	-	-	-
SB/SD/CS	-11.2	-	-	-	-	-	-9.4	-	-	-	-	-	-14.1	-	-	-	-	-
SB/MD/NS	0.0	-2.4	-7.4	-7.4	-7.9	-7.9	0.0	-4.0	-9.8	-9.8	-10.4	-10.4	-15.9	-17.3	-18.1	-18.1	-18.2	-18.2
SB/MD/CS	-11.2	-13.5	-19.0	-19.0	-19.5	-19.5	-9.4	-13.4	-19.1	-19.1	-19.6	-19.7	-14.1	-15.7	-15.9	-15.9	-15.8	-15.9
DB/SD/NS	-2.8	-	-	-	-	-	-2.2	-	-	-	-	-	-15.4	-	-	-	-	-
DB/SD/CS	-13.4	-	-	-	-	-	-11.3	-	-	-	-	-	-13.6	-	-	-	-	-
DB/MD/NS	-2.8	-5.4	-10.3	-10.3	-10.9	-10.9	-2.2	-6.2	-12.2	-12.2	-12.9	-12.9	-15.4	-16.6	-17.7	-17.7	-17.7	-17.7
DB/MD/CS	-13.4	-16.0	-21.4	-21.4	-22.0	-22.0	-11.3	-15.3	-21.2	-21.2	-21.7	-21.8	-13.6	-15.1	-15.4	-15.5	-15.3	-15.4

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 27 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 8$, $\gamma = 1.0$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-17.1	-	-	-	-	-
SB/SD/CS	-11.0	-	-	-	-	-	-9.4	-	-	-	-	-	-15.7	-	-	-	-	-
SB/MD/NS	0.0	-1.8	-4.6	-4.6	-5.0	-5.0	0.0	-2.5	-5.4	-5.4	-5.7	-5.7	-17.1	-17.8	-17.8	-17.8	-17.7	-17.7
SB/MD/CS	-11.0	-13.1	-15.9	-15.9	-16.3	-16.3	-9.4	-12.0	-14.6	-14.6	-14.8	-14.8	-15.7	-16.2	-15.9	-15.9	-15.7	-15.7
DB/SD/NS	-3.2	-	-	-	-	-	-2.9	-	-	-	-	-	-16.9	-	-	-	-	-
DB/SD/CS	-13.6	-	-	-	-	-	-11.7	-	-	-	-	-	-15.3	-	-	-	-	-
DB/MD/NS	-3.2	-5.0	-7.8	-7.8	-8.2	-8.2	-2.9	-5.5	-8.5	-8.5	-8.9	-8.9	-16.9	-17.6	-17.8	-17.8	-17.7	-17.8
DB/MD/CS	-13.6	-15.7	-18.5	-18.5	-19.0	-19.0	-11.7	-14.3	-16.9	-17.0	-17.4	-17.4	-15.3	-15.9	-15.6	-15.6	-15.5	-15.5

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 28 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 12$, $\gamma = 1.0$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-22.4	-	-	-	-	-
SB/SD/CS	-11.3	-	-	-	-	-	-8.9	-	-	-	-	-	-20.3	-	-	-	-	-
SB/MD/NS	0.0	-0.4	-1.8	-1.8	-2.0	-2.0	0.0	-1.3	-2.4	-2.5	-2.6	-2.7	-22.4	-23.2	-23.0	-23.0	-23.0	-23.0
SB/MD/CS	-11.3	-12.1	-13.1	-13.1	-13.3	-13.3	-8.9	-10.2	-11.5	-11.6	-11.7	-11.8	-20.3	-20.8	-21.0	-21.0	-21.0	-21.0
DB/SD/NS	-3.8	-	-	-	-	-	-3.1	-	-	-	-	-	-21.9	-	-	-	-	-
DB/SD/CS	-14.2	-	-	-	-	-	-11.3	-	-	-	-	-	-19.8	-	-	-	-	-
DB/MD/NS	-3.8	-4.3	-5.8	-5.8	-6.2	-6.2	-3.1	-4.5	-5.8	-5.8	-6.1	-6.1	-21.9	-22.6	-22.5	-22.5	-22.3	-22.3
DB/MD/CS	-14.2	-15.1	-16.2	-16.2	-16.5	-16.5	-11.3	-12.8	-14.1	-14.2	-14.4	-14.4	-19.8	-20.4	-20.5	-20.5	-20.4	-20.5

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 29 **Average policy and layout improvement potential with $\kappa = 2$, $\beta = 16$, $\gamma = 1.0$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.8	-	-	-	-	-
SB/SD/CS	-9.4	-	-	-	-	-	-7.9	-	-	-	-	-	-14.4	-	-	-	-	-
SB/MD/NS	0.0	-0.4	-1.1	-1.1	-1.2	-1.3	0.0	-0.7	-0.9	-1.1	-1.1	-1.2	-15.8	-16.0	-15.7	-15.8	-15.8	-15.8
SB/MD/CS	-9.4	-9.9	-10.2	-10.3	-10.3	-10.4	-7.9	-8.7	-8.9	-9.1	-9.2	-9.3	-14.4	-14.7	-14.7	-14.7	-14.8	-14.8
DB/SD/NS	-3.9	-	-	-	-	-	-3.3	-	-	-	-	-	-15.3	-	-	-	-	-
DB/SD/CS	-12.8	-	-	-	-	-	-10.6	-	-	-	-	-	-13.8	-	-	-	-	-
DB/MD/NS	-3.9	-4.4	-5.2	-5.3	-5.5	-5.6	-3.3	-4.0	-4.3	-4.4	-4.5	-4.6	-15.3	-15.4	-15.0	-15.1	-15.0	-14.9
DB/MD/CS	-12.8	-13.3	-13.7	-13.8	-13.9	-14.0	-10.6	-11.6	-11.7	-12.0	-12.0	-12.1	-13.8	-14.1	-13.9	-14.0	-14.0	-14.0

The table shows for $\kappa = 2$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

I.2. Results for cart capacity $\kappa = 3$

Table 30 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 2$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-1.8	-	-	-	-	-
SB/SD/CS	-3.5	-	-	-	-	-	-4.0	-	-	-	-	-	-2.3	-	-	-	-	-
SB/MD/NS	0.0	-5.9	-12.5	-12.7	-13.5	-13.6	0.0	-7.7	-15.0	-15.2	-16.0	-16.1	-1.8	-3.7	-4.7	-4.7	-4.8	-4.7
SB/MD/CS	-3.5	-9.4	-16.3	-16.5	-17.3	-17.4	-4.0	-11.6	-19.2	-19.3	-20.2	-20.3	-2.3	-4.2	-5.2	-5.2	-5.2	-5.2
DB/SD/NS	-4.0	-	-	-	-	-	-3.1	-	-	-	-	-	-1.0	-	-	-	-	-
DB/SD/CS	-6.9	-	-	-	-	-	-6.8	-	-	-	-	-	-1.8	-	-	-	-	-
DB/MD/NS	-4.0	-10.2	-16.4	-16.7	-17.3	-17.5	-3.1	-11.2	-17.7	-18.3	-18.7	-19.2	-1.0	-2.9	-3.3	-3.7	-3.4	-3.8
DB/MD/CS	-6.9	-13.1	-19.8	-20.0	-20.6	-20.7	-6.8	-14.6	-21.4	-22.0	-22.5	-22.9	-1.8	-3.5	-3.9	-4.3	-4.2	-4.5

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 31 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 4$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-4.0	-	-	-	-	-
SB/SD/CS	-4.9	-	-	-	-	-	-4.6	-	-	-	-	-	-3.7	-	-	-	-	-
SB/MD/NS	0.0	-2.9	-7.7	-7.8	-8.4	-8.4	0.0	-3.2	-9.3	-9.4	-10.0	-10.0	-4.0	-4.2	-5.7	-5.6	-5.6	-5.6
SB/MD/CS	-4.9	-7.7	-12.9	-13.1	-13.6	-13.7	-4.6	-7.5	-14.3	-14.4	-14.8	-14.8	-3.7	-3.9	-5.5	-5.4	-5.4	-5.4
DB/SD/NS	-5.9	-	-	-	-	-	-4.3	-	-	-	-	-	-2.2	-	-	-	-	-
DB/SD/CS	-9.8	-	-	-	-	-	-8.6	-	-	-	-	-	-2.7	-	-	-	-	-
DB/MD/NS	-5.9	-9.1	-14.0	-14.2	-14.7	-14.9	-4.3	-8.7	-14.6	-14.9	-15.3	-15.6	-2.2	-3.5	-4.5	-4.6	-4.5	-4.6
DB/MD/CS	-9.8	-12.9	-18.3	-18.5	-19.1	-19.3	-8.6	-12.1	-18.8	-18.9	-19.5	-19.6	-2.7	-3.1	-4.5	-4.4	-4.4	-4.3

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 32 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 6$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-3.0	-	-	-	-	-
SB/SD/CS	-3.5	-	-	-	-	-	-4.8	-	-	-	-	-	-4.3	-	-	-	-	-
SB/MD/NS	0.0	-1.9	-4.7	-4.8	-5.3	-5.3	0.0	-2.4	-5.8	-6.0	-6.3	-6.4	-3.0	-3.6	-4.1	-4.2	-4.0	-4.2
SB/MD/CS	-3.5	-5.4	-8.7	-8.8	-9.2	-9.3	-4.8	-6.9	-10.8	-11.0	-11.4	-11.5	-4.3	-4.6	-5.3	-5.4	-5.4	-5.4
DB/SD/NS	-5.4	-	-	-	-	-	-4.0	-	-	-	-	-	-1.6	-	-	-	-	-
DB/SD/CS	-8.6	-	-	-	-	-	-8.6	-	-	-	-	-	-3.0	-	-	-	-	-
DB/MD/NS	-5.4	-8.7	-11.1	-11.5	-11.7	-12.0	-4.0	-8.7	-10.9	-11.6	-11.6	-12.1	-1.6	-3.0	-2.8	-3.0	-2.8	-3.0
DB/MD/CS	-8.6	-11.3	-14.2	-14.5	-14.8	-15.0	-8.6	-12.1	-15.5	-15.9	-16.2	-16.4	-3.0	-3.8	-4.5	-4.5	-4.5	-4.6

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 33 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 8$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	3.7	-	-	-	-	-
SB/SD/CS	-2.7	-	-	-	-	-	-4.1	-	-	-	-	-	2.3	-	-	-	-	-
SB/MD/NS	0.0	-4.6	-4.4	-5.6	-5.1	-6.1	0.0	-3.6	-4.1	-5.1	-4.7	-5.5	3.7	4.8	4.1	4.4	4.3	4.4
SB/MD/CS	-2.7	-5.8	-6.8	-7.5	-7.5	-8.1	-4.1	-6.4	-7.7	-8.4	-8.3	-8.8	2.3	3.2	2.7	2.8	2.9	3.0
DB/SD/NS	-2.9	-	-	-	-	-	-3.7	-	-	-	-	-	2.8	-	-	-	-	-
DB/SD/CS	-5.8	-	-	-	-	-	-7.7	-	-	-	-	-	1.7	-	-	-	-	-
DB/MD/NS	-2.9	-9.1	-8.9	-10.6	-9.6	-11.0	-3.7	-8.1	-8.4	-9.5	-9.2	-9.9	2.8	5.0	4.3	5.1	4.3	5.1
DB/MD/CS	-5.8	-10.6	-10.9	-12.2	-11.7	-12.7	-7.7	-10.3	-11.6	-12.3	-12.2	-12.7	1.7	4.2	3.0	3.8	3.2	3.8

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 34 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 12$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-6.2	-	-	-	-	-
SB/SD/CS	-5.0	-	-	-	-	-	-5.4	-	-	-	-	-	-6.7	-	-	-	-	-
SB/MD/NS	0.0	-4.0	-9.1	-9.1	-9.9	-9.9	0.0	-5.5	-11.4	-11.6	-12.0	-12.2	-6.2	-7.7	-8.5	-8.7	-8.4	-8.5
SB/MD/CS	-5.0	-9.0	-14.4	-14.4	-15.2	-15.2	-5.4	-10.8	-16.8	-17.0	-17.5	-17.6	-6.7	-8.2	-8.8	-8.9	-8.7	-8.9
DB/SD/NS	-4.5	-	-	-	-	-	-3.5	-	-	-	-	-	-5.3	-	-	-	-	-
DB/SD/CS	-8.3	-	-	-	-	-	-8.2	-	-	-	-	-	-6.1	-	-	-	-	-
DB/MD/NS	-4.5	-8.7	-13.9	-14.0	-14.8	-14.8	-3.5	-9.2	-14.7	-15.0	-15.4	-15.6	-5.3	-6.8	-7.0	-7.3	-6.9	-7.1
DB/MD/CS	-8.3	-12.6	-17.9	-18.0	-18.8	-18.8	-8.2	-13.8	-19.4	-19.6	-20.2	-20.4	-6.1	-7.7	-8.0	-8.2	-8.0	-8.1

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 35 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 16$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-5.6	-	-	-	-	-
SB/SD/CS	-6.3	-	-	-	-	-	-5.5	-	-	-	-	-	-5.0	-	-	-	-	-
SB/MD/NS	0.0	-2.0	-5.3	-5.5	-5.9	-6.1	0.0	-2.2	-6.5	-6.6	-7.0	-7.0	-5.6	-5.8	-6.8	-6.6	-6.7	-6.5
SB/MD/CS	-6.3	-8.3	-12.2	-12.3	-12.7	-12.8	-5.5	-7.6	-12.3	-12.3	-12.7	-12.7	-5.0	-5.0	-5.9	-5.8	-5.8	-5.6
DB/SD/NS	-5.7	-	-	-	-	-	-4.3	-	-	-	-	-	-4.3	-	-	-	-	-
DB/SD/CS	-10.7	-	-	-	-	-	-9.7	-	-	-	-	-	-4.7	-	-	-	-	-
DB/MD/NS	-5.7	-8.0	-11.6	-12.0	-12.3	-12.6	-4.3	-6.8	-11.0	-11.1	-11.7	-11.8	-4.3	-4.4	-4.9	-4.6	-4.9	-4.6
DB/MD/CS	-10.7	-13.2	-17.0	-17.2	-17.6	-17.8	-9.7	-11.8	-16.2	-16.3	-16.8	-16.8	-4.7	-4.2	-4.8	-4.6	-4.7	-4.5

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 36 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 2$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-5.6	-	-	-	-	-
SB/SD/CS	-4.5	-	-	-	-	-	-5.5	-	-	-	-	-	-6.6	-	-	-	-	-
SB/MD/NS	0.0	-0.9	-3.4	-3.5	-3.8	-3.9	0.0	-1.5	-4.2	-4.3	-4.5	-4.6	-5.6	-6.3	-6.4	-6.5	-6.4	-6.4
SB/MD/CS	-4.5	-5.4	-8.3	-8.4	-8.7	-8.7	-5.5	-6.8	-9.8	-10.0	-10.0	-10.1	-6.6	-7.1	-7.2	-7.3	-7.1	-7.1
DB/SD/NS	-3.8	-	-	-	-	-	-3.7	-	-	-	-	-	-5.5	-	-	-	-	-
DB/SD/CS	-8.1	-	-	-	-	-	-8.9	-	-	-	-	-	-6.5	-	-	-	-	-
DB/MD/NS	-3.8	-5.2	-7.4	-7.5	-7.9	-8.0	-3.7	-5.4	-7.8	-8.0	-8.3	-8.4	-5.5	-5.8	-6.0	-6.1	-6.0	-6.1
DB/MD/CS	-8.1	-9.2	-12.0	-12.2	-12.3	-12.4	-8.9	-10.1	-13.1	-13.2	-13.3	-13.4	-6.5	-6.6	-6.8	-6.8	-6.7	-6.7

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 37 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 4$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	0.2	-	-	-	-	-
SB/SD/CS	-3.6	-	-	-	-	-	-4.7	-	-	-	-	-	-1.0	-	-	-	-	-
SB/MD/NS	0.0	-1.9	-3.3	-3.6	-3.8	-4.1	0.0	-1.3	-2.7	-3.0	-3.0	-3.2	0.2	0.8	0.9	0.9	1.1	1.1
SB/MD/CS	-3.6	-4.7	-6.4	-6.6	-6.8	-6.9	-4.7	-6.0	-7.3	-7.6	-7.5	-7.8	-1.0	-1.1	-0.7	-0.8	-0.6	-0.7
DB/SD/NS	-3.6	-	-	-	-	-	-3.4	-	-	-	-	-	0.5	-	-	-	-	-
DB/SD/CS	-6.9	-	-	-	-	-	-7.6	-	-	-	-	-	-0.4	-	-	-	-	-
DB/MD/NS	-3.6	-5.4	-7.0	-7.4	-7.5	-7.8	-3.4	-4.6	-6.0	-6.3	-6.3	-6.6	0.5	1.2	1.3	1.4	1.5	1.6
DB/MD/CS	-6.9	-7.8	-9.7	-9.9	-10.1	-10.3	-7.6	-8.7	-10.1	-10.3	-10.4	-10.6	-0.4	-0.7	-0.2	-0.2	-0.1	-0.2

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 38 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 6$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.3	-	-	-	-	-
SB/SD/CS	-6.5	-	-	-	-	-	-6.3	-	-	-	-	-	-9.1	-	-	-	-	-
SB/MD/NS	0.0	-3.1	-8.0	-8.0	-8.7	-8.7	0.0	-4.0	-9.9	-9.9	-10.4	-10.5	-9.3	-10.1	-11.0	-11.0	-10.9	-10.9
SB/MD/CS	-6.5	-9.6	-14.7	-14.7	-15.3	-15.3	-6.3	-10.1	-16.1	-16.1	-16.7	-16.7	-9.1	-9.9	-10.8	-10.8	-10.7	-10.7
DB/SD/NS	-4.1	-	-	-	-	-	-4.1	-	-	-	-	-	-9.3	-	-	-	-	-
DB/SD/CS	-9.7	-	-	-	-	-	-9.7	-	-	-	-	-	-9.3	-	-	-	-	-
DB/MD/NS	-4.1	-7.2	-12.3	-12.3	-12.9	-12.9	-4.1	-7.8	-13.9	-13.9	-14.4	-14.4	-9.3	-10.0	-10.8	-10.9	-10.7	-10.8
DB/MD/CS	-9.7	-12.7	-17.9	-17.9	-18.5	-18.5	-9.7	-13.2	-19.4	-19.4	-20.0	-20.0	-9.3	-9.9	-11.0	-11.0	-10.9	-11.0

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 39 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 8$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-8.8	-	-	-	-	-
SB/SD/CS	-7.5	-	-	-	-	-	-6.2	-	-	-	-	-	-7.6	-	-	-	-	-
SB/MD/NS	0.0	-1.7	-4.6	-4.6	-5.2	-5.2	0.0	-2.4	-5.7	-5.7	-6.2	-6.2	-8.8	-9.4	-9.9	-9.9	-9.8	-9.7
SB/MD/CS	-7.5	-9.3	-12.5	-12.5	-13.1	-13.1	-6.2	-8.5	-12.1	-12.1	-12.5	-12.5	-7.6	-8.1	-8.4	-8.4	-8.2	-8.2
DB/SD/NS	-4.6	-	-	-	-	-	-3.6	-	-	-	-	-	-7.8	-	-	-	-	-
DB/SD/CS	-11.1	-	-	-	-	-	-9.1	-	-	-	-	-	-6.9	-	-	-	-	-
DB/MD/NS	-4.6	-6.9	-9.4	-9.6	-10.0	-10.1	-3.6	-5.9	-9.4	-9.4	-9.9	-9.9	-7.8	-7.8	-8.7	-8.5	-8.6	-8.5
DB/MD/CS	-11.1	-13.1	-16.2	-16.2	-16.9	-16.9	-9.1	-11.5	-15.1	-15.1	-15.6	-15.6	-6.9	-7.3	-7.7	-7.6	-7.5	-7.4

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 40 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 12$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.5	-	-	-	-	-
SB/SD/CS	-6.1	-	-	-	-	-	-6.4	-	-	-	-	-	-9.9	-	-	-	-	-
SB/MD/NS	0.0	-0.7	-2.3	-2.3	-2.8	-2.8	0.0	-1.4	-3.2	-3.3	-3.5	-3.6	-9.5	-10.2	-10.4	-10.5	-10.2	-10.2
SB/MD/CS	-6.1	-6.8	-8.6	-8.6	-9.1	-9.1	-6.4	-7.9	-9.7	-9.8	-10.0	-10.0	-9.9	-10.6	-10.6	-10.7	-10.4	-10.5
DB/SD/NS	-3.3	-	-	-	-	-	-3.3	-	-	-	-	-	-9.4	-	-	-	-	-
DB/SD/CS	-9.0	-	-	-	-	-	-9.2	-	-	-	-	-	-9.6	-	-	-	-	-
DB/MD/NS	-3.3	-4.2	-5.7	-5.7	-6.2	-6.2	-3.3	-4.8	-6.6	-6.7	-6.9	-6.9	-9.4	-10.2	-10.4	-10.5	-10.1	-10.2
DB/MD/CS	-9.0	-9.9	-11.6	-11.6	-12.1	-12.1	-9.2	-10.8	-12.4	-12.6	-12.7	-12.7	-9.6	-10.4	-10.4	-10.5	-10.1	-10.1

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 41 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 16$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-2.4	-	-	-	-	-
SB/SD/CS	-4.2	-	-	-	-	-	-5.7	-	-	-	-	-	-3.9	-	-	-	-	-
SB/MD/NS	0.0	-0.7	-2.5	-2.6	-2.9	-2.9	0.0	-0.9	-2.1	-2.1	-2.3	-2.4	-2.4	-2.6	-1.9	-2.0	-1.8	-1.8
SB/MD/CS	-4.2	-4.8	-6.3	-6.4	-6.6	-6.6	-5.7	-6.8	-7.8	-7.9	-8.0	-8.0	-3.9	-4.4	-4.0	-4.0	-3.8	-3.9
DB/SD/NS	-2.9	-	-	-	-	-	-3.1	-	-	-	-	-	-2.6	-	-	-	-	-
DB/SD/CS	-6.9	-	-	-	-	-	-8.3	-	-	-	-	-	-3.9	-	-	-	-	-
DB/MD/NS	-2.9	-3.6	-5.5	-5.5	-5.8	-5.8	-3.1	-3.9	-5.2	-5.3	-5.5	-5.5	-2.6	-2.6	-2.0	-2.1	-2.0	-2.1
DB/MD/CS	-6.9	-7.6	-9.1	-9.2	-9.4	-9.4	-8.3	-9.4	-10.4	-10.6	-10.5	-10.7	-3.9	-4.3	-3.8	-3.9	-3.6	-3.7

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 42 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 2$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.1	-	-	-	-	-
SB/SD/CS	-8.1	-	-	-	-	-	-7.1	-	-	-	-	-	-10.1	-	-	-	-	-
SB/MD/NS	0.0	-2.4	-7.0	-7.0	-7.6	-7.6	0.0	-3.5	-8.6	-8.7	-9.1	-9.1	-11.1	-12.1	-12.7	-12.7	-12.6	-12.6
SB/MD/CS	-8.1	-10.6	-15.4	-15.4	-16.0	-16.0	-7.1	-10.6	-15.7	-15.8	-16.2	-16.2	-10.1	-11.1	-11.4	-11.5	-11.3	-11.4
DB/SD/NS	-3.8	-	-	-	-	-	-3.8	-	-	-	-	-	-11.2	-	-	-	-	-
DB/SD/CS	-10.9	-	-	-	-	-	-10.1	-	-	-	-	-	-10.3	-	-	-	-	-
DB/MD/NS	-3.8	-6.4	-10.9	-10.9	-11.6	-11.6	-3.8	-7.4	-12.7	-12.7	-13.2	-13.2	-11.2	-12.2	-12.8	-12.9	-12.7	-12.8
DB/MD/CS	-10.9	-13.3	-18.2	-18.2	-18.8	-18.8	-10.1	-13.6	-19.0	-19.0	-19.6	-19.6	-10.3	-11.5	-11.9	-11.9	-11.9	-12.0

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 43 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 4$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.9	-	-	-	-	-
SB/SD/CS	-8.7	-	-	-	-	-	-6.8	-	-	-	-	-	-8.1	-	-	-	-	-
SB/MD/NS	0.0	-1.5	-3.9	-4.0	-4.3	-4.3	0.0	-2.3	-5.1	-5.1	-5.4	-5.4	-9.9	-10.6	-11.0	-11.0	-10.9	-10.9
SB/MD/CS	-8.7	-10.4	-13.1	-13.1	-13.5	-13.5	-6.8	-9.2	-11.8	-11.8	-12.0	-12.0	-8.1	-9.0	-8.7	-8.7	-8.6	-8.6
DB/SD/NS	-4.1	-	-	-	-	-	-3.2	-	-	-	-	-	-9.1	-	-	-	-	-
DB/SD/CS	-11.8	-	-	-	-	-	-9.5	-	-	-	-	-	-7.7	-	-	-	-	-
DB/MD/NS	-4.1	-5.8	-8.1	-8.2	-8.5	-8.5	-3.2	-5.5	-8.5	-8.5	-8.7	-8.8	-9.1	-9.7	-10.2	-10.2	-10.1	-10.1
DB/MD/CS	-11.8	-13.5	-16.2	-16.2	-16.6	-16.6	-9.5	-11.9	-14.6	-14.6	-14.8	-14.8	-7.7	-8.4	-8.4	-8.4	-8.2	-8.2

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 44 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 6$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-12.6	-	-	-	-	-
SB/SD/CS	-7.1	-	-	-	-	-	-7.3	-	-	-	-	-	-12.9	-	-	-	-	-
SB/MD/NS	0.0	-0.4	-1.9	-1.9	-2.1	-2.1	0.0	-1.2	-2.4	-2.5	-2.6	-2.6	-12.6	-13.4	-13.1	-13.2	-13.0	-13.1
SB/MD/CS	-7.1	-7.6	-9.1	-9.1	-9.3	-9.3	-7.3	-8.7	-9.8	-9.9	-9.9	-9.9	-12.9	-13.7	-13.3	-13.4	-13.2	-13.2
DB/SD/NS	-3.6	-	-	-	-	-	-3.6	-	-	-	-	-	-12.6	-	-	-	-	-
DB/SD/CS	-10.2	-	-	-	-	-	-10.6	-	-	-	-	-	-13.0	-	-	-	-	-
DB/MD/NS	-3.6	-4.1	-5.6	-5.6	-5.8	-5.8	-3.6	-4.9	-5.8	-5.9	-6.1	-6.1	-12.6	-13.4	-12.9	-13.0	-12.9	-13.0
DB/MD/CS	-10.2	-10.8	-12.1	-12.1	-12.4	-12.4	-10.6	-12.0	-12.9	-13.0	-13.2	-13.2	-13.0	-13.8	-13.4	-13.5	-13.4	-13.4

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 45 **Average policy and layout improvement potential with $\kappa = 3$, $\beta = 8$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-5.3	-	-	-	-	-
SB/SD/CS	-5.4	-	-	-	-	-	-6.5	-	-	-	-	-	-6.4	-	-	-	-	-
SB/MD/NS	0.0	-0.3	-1.6	-1.6	-1.9	-1.9	0.0	-0.5	-1.1	-1.1	-1.1	-1.2	-5.3	-5.4	-4.7	-4.8	-4.6	-4.6
SB/MD/CS	-5.4	-5.8	-6.5	-6.6	-6.7	-6.7	-6.5	-7.3	-7.5	-7.7	-7.6	-7.8	-6.4	-6.8	-6.3	-6.4	-6.2	-6.3
DB/SD/NS	-2.7	-	-	-	-	-	-2.7	-	-	-	-	-	-5.2	-	-	-	-	-
DB/SD/CS	-7.9	-	-	-	-	-	-8.6	-	-	-	-	-	-6.0	-	-	-	-	-
DB/MD/NS	-2.7	-3.2	-4.3	-4.4	-4.5	-4.6	-2.7	-3.4	-3.9	-4.0	-4.0	-4.1	-5.2	-5.4	-4.9	-4.9	-4.8	-4.8
DB/MD/CS	-7.9	-8.4	-9.0	-9.1	-9.2	-9.2	-8.6	-9.5	-9.8	-10.0	-10.1	-10.2	-6.0	-6.4	-6.1	-6.2	-6.2	-6.2

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 46 **Average policy and layout improvement potential with $\kappa = 3$, $\beta = 12$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-13.3	-	-	-	-	-
SB/SD/CS	-9.1	-	-	-	-	-	-7.8	-	-	-	-	-	-12.1	-	-	-	-	-
SB/MD/NS	0.0	-2.1	-5.7	-5.7	-6.4	-6.4	0.0	-2.9	-7.4	-7.4	-8.1	-8.1	-13.3	-13.9	-14.8	-14.8	-14.7	-14.7
SB/MD/CS	-9.1	-11.2	-15.1	-15.1	-15.8	-15.8	-7.8	-10.8	-15.2	-15.2	-15.8	-15.8	-12.1	-12.8	-13.3	-13.3	-13.2	-13.2
DB/SD/NS	-2.6	-	-	-	-	-	-2.8	-	-	-	-	-	-13.5	-	-	-	-	-
DB/SD/CS	-11.0	-	-	-	-	-	-10.0	-	-	-	-	-	-12.3	-	-	-	-	-
DB/MD/NS	-2.6	-4.7	-8.3	-8.3	-9.0	-9.0	-2.8	-5.7	-10.2	-10.2	-10.9	-10.9	-13.5	-14.2	-15.0	-15.1	-15.1	-15.1
DB/MD/CS	-11.0	-13.2	-17.1	-17.1	-17.7	-17.7	-10.0	-13.1	-17.5	-17.5	-18.1	-18.1	-12.3	-13.2	-13.6	-13.6	-13.6	-13.6

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 47 **Average policy and layout improvement potential with $\kappa = 3$, $\beta = 16$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-14.0	-	-	-	-	-
SB/SD/CS	-10.1	-	-	-	-	-	-8.4	-	-	-	-	-	-12.5	-	-	-	-	-
SB/MD/NS	0.0	-0.7	-3.0	-3.0	-3.5	-3.5	0.0	-1.3	-3.7	-3.7	-4.0	-4.0	-14.0	-14.6	-14.6	-14.6	-14.5	-14.5
SB/MD/CS	-10.1	-11.0	-13.4	-13.4	-13.9	-13.9	-8.4	-9.7	-11.8	-11.8	-12.0	-12.0	-12.5	-12.9	-12.6	-12.6	-12.2	-12.2
DB/SD/NS	-4.2	-	-	-	-	-	-3.8	-	-	-	-	-	-13.6	-	-	-	-	-
DB/SD/CS	-13.3	-	-	-	-	-	-11.6	-	-	-	-	-	-12.4	-	-	-	-	-
DB/MD/NS	-4.2	-4.8	-7.4	-7.4	-7.9	-7.9	-3.8	-5.1	-7.2	-7.2	-7.6	-7.6	-13.6	-14.2	-13.8	-13.8	-13.7	-13.7
DB/MD/CS	-13.3	-14.1	-16.6	-16.6	-17.1	-17.1	-11.6	-12.9	-14.8	-14.9	-15.1	-15.1	-12.4	-12.9	-12.3	-12.3	-12.1	-12.1

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 48 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 2$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-17.7	-	-	-	-	-
SB/SD/CS	-9.1	-	-	-	-	-	-8.7	-	-	-	-	-	-17.3	-	-	-	-	-
SB/MD/NS	0.0	-0.2	-1.4	-1.4	-1.7	-1.7	0.0	-0.8	-1.5	-1.6	-1.7	-1.7	-17.7	-18.2	-17.8	-17.8	-17.7	-17.7
SB/MD/CS	-9.1	-9.4	-10.3	-10.3	-10.6	-10.6	-8.7	-9.5	-10.3	-10.4	-10.5	-10.5	-17.3	-17.8	-17.7	-17.7	-17.6	-17.6
DB/SD/NS	-3.0	-	-	-	-	-	-3.3	-	-	-	-	-	-18.0	-	-	-	-	-
DB/SD/CS	-11.7	-	-	-	-	-	-11.5	-	-	-	-	-	-17.5	-	-	-	-	-
DB/MD/NS	-3.0	-3.4	-4.4	-4.4	-4.7	-4.7	-3.3	-4.0	-4.8	-4.8	-5.0	-5.0	-18.0	-18.2	-18.0	-18.0	-17.9	-17.9
DB/MD/CS	-11.7	-12.1	-12.9	-12.9	-13.2	-13.2	-11.5	-12.4	-13.1	-13.2	-13.2	-13.3	-17.5	-17.9	-17.8	-17.9	-17.7	-17.8

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 49 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 4$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.7	-	-	-	-	-
SB/SD/CS	-7.2	-	-	-	-	-	-7.6	-	-	-	-	-	-10.1	-	-	-	-	-
SB/MD/NS	0.0	-0.2	-1.0	-1.0	-1.2	-1.2	0.0	-0.5	-0.6	-0.7	-0.7	-0.8	-9.7	-9.9	-9.2	-9.4	-9.2	-9.3
SB/MD/CS	-7.2	-7.4	-7.7	-7.8	-7.8	-7.9	-7.6	-8.3	-8.3	-8.4	-8.4	-8.5	-10.1	-10.5	-10.2	-10.3	-10.2	-10.3
DB/SD/NS	-2.8	-	-	-	-	-	-3.2	-	-	-	-	-	-10.0	-	-	-	-	-
DB/SD/CS	-9.8	-	-	-	-	-	-10.3	-	-	-	-	-	-10.1	-	-	-	-	-
DB/MD/NS	-2.8	-3.1	-3.8	-3.8	-4.0	-4.0	-3.2	-3.9	-3.9	-4.1	-4.1	-4.2	-10.0	-10.4	-9.8	-10.0	-9.8	-9.9
DB/MD/CS	-9.8	-10.2	-10.4	-10.5	-10.5	-10.6	-10.3	-11.1	-11.1	-11.3	-11.3	-11.4	-10.1	-10.5	-10.4	-10.5	-10.4	-10.5

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 50 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 6$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-14.1	-	-	-	-	-
SB/SD/CS	-10.3	-	-	-	-	-	-8.9	-	-	-	-	-	-12.7	-	-	-	-	-
SB/MD/NS	0.0	-1.4	-4.7	-4.7	-5.3	-5.3	0.0	-2.4	-6.3	-6.3	-6.8	-6.8	-14.1	-15.0	-15.5	-15.5	-15.5	-15.5
SB/MD/CS	-10.3	-11.8	-15.3	-15.3	-15.8	-15.8	-8.9	-11.4	-15.2	-15.2	-15.6	-15.6	-12.7	-13.7	-13.8	-13.8	-13.7	-13.7
DB/SD/NS	-2.3	-	-	-	-	-	-2.7	-	-	-	-	-	-14.4	-	-	-	-	-
DB/SD/CS	-12.4	-	-	-	-	-	-11.3	-	-	-	-	-	-12.9	-	-	-	-	-
DB/MD/NS	-2.3	-3.7	-7.1	-7.1	-7.6	-7.6	-2.7	-5.1	-9.1	-9.1	-9.7	-9.7	-14.4	-15.4	-16.0	-16.0	-16.0	-16.0
DB/MD/CS	-12.4	-13.8	-17.4	-17.4	-18.0	-18.0	-11.3	-13.8	-17.7	-17.7	-18.1	-18.1	-12.9	-14.0	-14.1	-14.1	-14.0	-14.0

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 51 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 8$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.6	-	-	-	-	-
SB/SD/CS	-10.4	-	-	-	-	-	-9.1	-	-	-	-	-	-14.4	-	-	-	-	-
SB/MD/NS	0.0	-0.9	-2.9	-2.9	-3.3	-3.3	0.0	-1.5	-3.4	-3.4	-3.6	-3.6	-15.6	-16.2	-16.1	-16.1	-15.9	-15.9
SB/MD/CS	-10.4	-11.6	-13.5	-13.5	-13.9	-13.9	-9.1	-10.7	-12.2	-12.2	-12.4	-12.4	-14.4	-14.9	-14.4	-14.4	-14.3	-14.3
DB/SD/NS	-3.3	-	-	-	-	-	-3.0	-	-	-	-	-	-15.3	-	-	-	-	-
DB/SD/CS	-13.0	-	-	-	-	-	-11.7	-	-	-	-	-	-14.4	-	-	-	-	-
DB/MD/NS	-3.3	-4.2	-6.2	-6.2	-6.6	-6.6	-3.0	-4.5	-6.6	-6.6	-6.9	-6.9	-15.3	-15.9	-15.9	-15.9	-15.8	-15.8
DB/MD/CS	-13.0	-14.1	-16.2	-16.2	-16.6	-16.6	-11.7	-13.3	-14.8	-14.8	-15.1	-15.1	-14.4	-14.9	-14.3	-14.3	-14.2	-14.2

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 52 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 12$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-20.3	-	-	-	-	-
SB/SD/CS	-10.2	-	-	-	-	-	-9.0	-	-	-	-	-	-19.1	-	-	-	-	-
SB/MD/NS	0.0	-0.1	-1.2	-1.2	-1.4	-1.4	0.0	-0.8	-1.6	-1.6	-1.7	-1.7	-20.3	-20.8	-20.6	-20.6	-20.5	-20.5
SB/MD/CS	-10.2	-10.6	-11.5	-11.5	-11.7	-11.7	-9.0	-9.8	-10.6	-10.6	-10.8	-10.8	-19.1	-19.6	-19.5	-19.5	-19.5	-19.5
DB/SD/NS	-3.2	-	-	-	-	-	-2.8	-	-	-	-	-	-20.0	-	-	-	-	-
DB/SD/CS	-12.9	-	-	-	-	-	-11.3	-	-	-	-	-	-18.7	-	-	-	-	-
DB/MD/NS	-3.2	-3.5	-4.6	-4.6	-4.8	-4.8	-2.8	-3.6	-4.5	-4.5	-4.7	-4.7	-20.0	-20.4	-20.2	-20.2	-20.2	-20.2
DB/MD/CS	-12.9	-13.3	-14.3	-14.3	-14.5	-14.5	-11.3	-12.2	-13.0	-13.1	-13.2	-13.2	-18.7	-19.2	-19.1	-19.1	-19.1	-19.1

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 53 Average policy and layout improvement potential with $\kappa = 3$, $\beta = 16$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-13.1	-	-	-	-	-
SB/SD/CS	-8.5	-	-	-	-	-	-8.1	-	-	-	-	-	-12.7	-	-	-	-	-
SB/MD/NS	0.0	-0.3	-0.8	-0.8	-1.0	-1.0	0.0	-0.5	-0.6	-0.7	-0.7	-0.8	-13.1	-13.3	-12.9	-13.0	-12.9	-12.9
SB/MD/CS	-8.5	-8.8	-9.0	-9.1	-9.2	-9.2	-8.1	-8.7	-8.7	-8.8	-8.9	-8.9	-12.7	-13.0	-12.8	-12.8	-12.8	-12.8
DB/SD/NS	-3.5	-	-	-	-	-	-3.5	-	-	-	-	-	-13.1	-	-	-	-	-
DB/SD/CS	-11.4	-	-	-	-	-	-10.9	-	-	-	-	-	-12.6	-	-	-	-	-
DB/MD/NS	-3.5	-3.9	-4.5	-4.6	-4.7	-4.7	-3.5	-4.0	-4.1	-4.2	-4.3	-4.3	-13.1	-13.1	-12.7	-12.8	-12.6	-12.7
DB/MD/CS	-11.4	-11.8	-12.0	-12.1	-12.2	-12.2	-10.9	-11.4	-11.5	-11.6	-11.7	-11.7	-12.6	-12.7	-12.6	-12.6	-12.6	-12.6

The table shows for $\kappa = 3$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

I.3. Results for cart capacity $\kappa = 4$

Table 54 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 2$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-1.6	-	-	-	-	-
SB/SD/CS	-3.4	-	-	-	-	-	-4.0	-	-	-	-	-	-2.2	-	-	-	-	-
SB/MD/NS	0.0	-4.8	-12.2	-12.3	-13.2	-13.2	0.0	-7.2	-14.4	-14.9	-15.3	-15.7	-1.6	-4.3	-4.2	-4.7	-4.1	-4.5
SB/MD/CS	-3.4	-8.3	-16.0	-16.1	-17.0	-17.1	-4.0	-10.9	-18.5	-19.0	-19.4	-19.8	-2.2	-4.6	-4.6	-5.0	-4.5	-4.9
DB/SD/NS	-6.4	-	-	-	-	-	-5.2	-	-	-	-	-	-0.3	-	-	-	-	-
DB/SD/CS	-9.1	-	-	-	-	-	-8.7	-	-	-	-	-	-1.2	-	-	-	-	-
DB/MD/NS	-6.4	-12.1	-18.8	-18.9	-19.6	-19.6	-5.2	-13.0	-19.4	-20.0	-20.2	-20.7	-0.3	-2.8	-2.3	-2.9	-2.3	-2.9
DB/MD/CS	-9.1	-14.7	-21.9	-22.0	-22.6	-22.6	-8.7	-16.1	-23.0	-23.5	-23.7	-24.1	-1.2	-3.3	-2.9	-3.5	-3.0	-3.5

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 55 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 4$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-3.3	-	-	-	-	-
SB/SD/CS	-5.0	-	-	-	-	-	-4.8	-	-	-	-	-	-3.1	-	-	-	-	-
SB/MD/NS	0.0	-4.1	-8.2	-8.7	-9.0	-9.4	0.0	-4.0	-9.7	-10.0	-10.5	-10.7	-3.3	-3.1	-4.9	-4.5	-4.8	-4.5
SB/MD/CS	-5.0	-8.5	-13.4	-13.7	-14.1	-14.3	-4.8	-7.9	-14.6	-14.8	-15.4	-15.4	-3.1	-2.7	-4.7	-4.5	-4.7	-4.6
DB/SD/NS	-8.4	-	-	-	-	-	-7.6	-	-	-	-	-	-2.3	-	-	-	-	-
DB/SD/CS	-12.0	-	-	-	-	-	-11.6	-	-	-	-	-	-2.8	-	-	-	-	-
DB/MD/NS	-8.4	-12.8	-16.6	-17.2	-17.6	-18.1	-7.6	-12.1	-17.7	-18.1	-18.2	-18.6	-2.3	-2.5	-4.3	-4.1	-3.9	-3.7
DB/MD/CS	-12.0	-15.9	-20.2	-20.9	-21.1	-21.5	-11.6	-15.0	-21.3	-21.6	-22.0	-22.2	-2.8	-2.1	-4.4	-3.9	-4.3	-3.9

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 56 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 6$, $\gamma = 0.25$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-1.7	-	-	-	-	-
SB/SD/CS	-3.8	-	-	-	-	-	-4.6	-	-	-	-	-	-2.6	-	-	-	-	-
SB/MD/NS	0.0	-1.8	-5.0	-5.3	-5.6	-5.8	0.0	-2.9	-6.2	-6.4	-6.8	-7.0	-1.7	-2.9	-3.0	-3.0	-3.1	-3.0
SB/MD/CS	-3.8	-5.2	-9.0	-9.1	-9.6	-9.7	-4.6	-7.0	-11.0	-11.1	-11.6	-11.7	-2.6	-3.6	-3.9	-4.0	-3.9	-4.0
DB/SD/NS	-7.6	-	-	-	-	-	-6.8	-	-	-	-	-	-0.7	-	-	-	-	-
DB/SD/CS	-10.0	-	-	-	-	-	-10.7	-	-	-	-	-	-2.3	-	-	-	-	-
DB/MD/NS	-7.6	-11.6	-13.7	-14.4	-14.2	-14.9	-6.8	-11.6	-13.6	-14.5	-14.2	-14.9	-0.7	-1.6	-1.5	-1.7	-1.5	-1.6
DB/MD/CS	-10.0	-13.5	-16.2	-16.9	-16.9	-17.5	-10.7	-14.2	-17.2	-17.8	-17.9	-18.4	-2.3	-2.4	-2.7	-2.7	-2.8	-2.7

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 57 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 8$, $\gamma = 0.25$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	2.5	-	-	-	-	-
SB/SD/CS	-3.5	-	-	-	-	-	-3.3	-	-	-	-	-	2.7	-	-	-	-	-
SB/MD/NS	0.0	-3.8	-4.2	-5.4	-4.8	-5.8	0.0	-2.1	-3.6	-4.2	-4.1	-4.5	2.5	4.3	3.1	3.8	3.2	3.8
SB/MD/CS	-3.5	-5.7	-7.2	-7.8	-7.7	-8.2	-3.3	-5.3	-7.0	-7.6	-7.5	-8.0	2.7	2.9	2.6	2.7	2.6	2.6
DB/SD/NS	-7.4	-	-	-	-	-	-7.0	-	-	-	-	-	2.9	-	-	-	-	-
DB/SD/CS	-10.3	-	-	-	-	-	-9.8	-	-	-	-	-	3.0	-	-	-	-	-
DB/MD/NS	-7.4	-13.1	-12.6	-14.5	-13.4	-14.9	-7.0	-11.1	-11.5	-12.7	-12.1	-13.0	2.9	5.1	4.0	4.8	4.1	5.0
DB/MD/CS	-10.3	-14.5	-15.0	-16.3	-15.7	-16.7	-9.8	-12.5	-13.7	-14.5	-14.4	-14.9	3.0	5.1	4.0	4.9	4.1	4.9

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 58 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 12$, $\gamma = 0.25$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-4.9	-	-	-	-	-
SB/SD/CS	-4.9	-	-	-	-	-	-5.5	-	-	-	-	-	-5.4	-	-	-	-	-
SB/MD/NS	0.0	-3.5	-9.2	-9.3	-9.9	-9.9	0.0	-5.1	-11.0	-11.4	-11.7	-12.0	-4.9	-6.7	-6.8	-7.2	-6.8	-7.2
SB/MD/CS	-4.9	-8.5	-14.5	-14.6	-15.2	-15.2	-5.5	-10.5	-16.5	-16.9	-17.3	-17.7	-5.4	-7.1	-7.2	-7.6	-7.4	-7.7
DB/SD/NS	-7.1	-	-	-	-	-	-6.6	-	-	-	-	-	-4.5	-	-	-	-	-
DB/SD/CS	-10.9	-	-	-	-	-	-10.9	-	-	-	-	-	-4.9	-	-	-	-	-
DB/MD/NS	-7.1	-11.3	-16.8	-17.0	-17.6	-17.8	-6.6	-12.2	-17.6	-18.1	-18.4	-18.7	-4.5	-6.0	-5.8	-6.1	-5.7	-6.0
DB/MD/CS	-10.9	-14.9	-20.6	-20.7	-21.4	-21.4	-10.9	-16.2	-21.6	-22.1	-22.3	-22.7	-4.9	-6.5	-6.2	-6.6	-6.1	-6.5

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 59 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 16$, $\gamma = 0.25$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-6.2	-	-	-	-	-
SB/SD/CS	-6.1	-	-	-	-	-	-5.4	-	-	-	-	-	-5.5	-	-	-	-	-
SB/MD/NS	0.0	-2.1	-5.9	-6.0	-6.4	-6.4	0.0	-2.2	-7.1	-7.1	-7.5	-7.5	-6.2	-6.2	-7.3	-7.2	-7.2	-7.2
SB/MD/CS	-6.1	-8.3	-12.5	-12.5	-13.0	-13.0	-5.4	-7.5	-12.8	-12.8	-13.2	-13.2	-5.5	-5.5	-6.5	-6.5	-6.5	-6.5
DB/SD/NS	-9.2	-	-	-	-	-	-6.4	-	-	-	-	-	-3.2	-	-	-	-	-
DB/SD/CS	-13.1	-	-	-	-	-	-11.3	-	-	-	-	-	-4.2	-	-	-	-	-
DB/MD/NS	-9.2	-11.6	-14.9	-15.3	-15.5	-15.8	-6.4	-9.4	-13.6	-13.8	-14.0	-14.1	-3.2	-3.6	-4.4	-4.2	-4.2	-4.0
DB/MD/CS	-13.1	-15.5	-19.2	-19.5	-19.7	-20.0	-11.3	-13.8	-18.2	-18.3	-18.5	-18.6	-4.2	-4.2	-4.8	-4.5	-4.5	-4.3

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 60 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 2$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-5.9	-	-	-	-	-
SB/SD/CS	-4.9	-	-	-	-	-	-5.4	-	-	-	-	-	-6.5	-	-	-	-	-
SB/MD/NS	0.0	-0.8	-3.5	-3.6	-3.9	-4.0	0.0	-1.8	-4.5	-4.7	-4.9	-5.0	-5.9	-6.9	-6.9	-7.0	-6.9	-7.0
SB/MD/CS	-4.9	-5.7	-8.7	-8.7	-9.1	-9.1	-5.4	-7.1	-10.3	-10.4	-10.6	-10.7	-6.5	-7.4	-7.6	-7.7	-7.5	-7.6
DB/SD/NS	-8.3	-	-	-	-	-	-7.5	-	-	-	-	-	-5.0	-	-	-	-	-
DB/SD/CS	-11.7	-	-	-	-	-	-11.6	-	-	-	-	-	-5.7	-	-	-	-	-
DB/MD/NS	-8.3	-9.8	-11.9	-12.1	-12.5	-12.6	-7.5	-9.1	-11.5	-11.6	-12.0	-12.1	-5.0	-5.1	-5.4	-5.3	-5.3	-5.2
DB/MD/CS	-11.7	-13.1	-15.4	-15.6	-16.0	-16.1	-11.6	-13.2	-15.9	-16.1	-16.4	-16.5	-5.7	-6.0	-6.5	-6.4	-6.4	-6.3

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 61 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 4$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-0.2	-	-	-	-	-
SB/SD/CS	-3.6	-	-	-	-	-	-4.2	-	-	-	-	-	-0.9	-	-	-	-	-
SB/MD/NS	0.0	-1.2	-3.0	-3.2	-3.4	-3.5	0.0	-1.5	-3.0	-3.2	-3.3	-3.5	-0.2	-0.5	-0.2	-0.2	-0.1	-0.2
SB/MD/CS	-3.6	-4.8	-6.6	-6.7	-6.9	-7.0	-4.2	-5.8	-7.3	-7.5	-7.6	-7.7	-0.9	-1.3	-1.0	-1.1	-1.0	-1.0
DB/SD/NS	-7.1	-	-	-	-	-	-5.9	-	-	-	-	-	1.0	-	-	-	-	-
DB/SD/CS	-9.4	-	-	-	-	-	-9.4	-	-	-	-	-	-0.1	-	-	-	-	-
DB/MD/NS	-7.1	-8.9	-10.5	-10.9	-10.8	-11.2	-5.9	-7.3	-8.9	-9.1	-9.2	-9.4	1.0	1.6	1.6	1.9	1.6	1.9
DB/MD/CS	-9.4	-11.3	-12.7	-13.1	-13.2	-13.6	-9.4	-10.8	-12.3	-12.5	-12.6	-12.7	-0.1	0.4	0.3	0.6	0.6	0.9

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 62 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 6$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.1	-	-	-	-	-
SB/SD/CS	-6.7	-	-	-	-	-	-5.9	-	-	-	-	-	-10.4	-	-	-	-	-
SB/MD/NS	0.0	-3.4	-7.9	-7.9	-8.5	-8.5	0.0	-4.2	-9.9	-10.0	-10.5	-10.5	-11.1	-12.0	-13.1	-13.1	-13.0	-13.1
SB/MD/CS	-6.7	-10.1	-15.0	-15.0	-15.6	-15.6	-5.9	-10.0	-15.9	-16.0	-16.5	-16.6	-10.4	-11.2	-12.2	-12.3	-12.1	-12.2
DB/SD/NS	-7.0	-	-	-	-	-	-5.5	-	-	-	-	-	-9.7	-	-	-	-	-
DB/SD/CS	-11.8	-	-	-	-	-	-10.2	-	-	-	-	-	-9.5	-	-	-	-	-
DB/MD/NS	-7.0	-10.1	-15.0	-15.1	-15.7	-15.7	-5.5	-9.6	-15.5	-15.6	-16.0	-16.1	-9.7	-10.6	-11.5	-11.5	-11.3	-11.4
DB/MD/CS	-11.8	-14.9	-20.1	-20.1	-20.7	-20.7	-10.2	-14.1	-20.5	-20.5	-20.9	-21.0	-9.5	-10.3	-11.5	-11.5	-11.4	-11.4

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 63 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 8$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-8.2	-	-	-	-	-
SB/SD/CS	-7.3	-	-	-	-	-	-6.2	-	-	-	-	-	-7.2	-	-	-	-	-
SB/MD/NS	0.0	-1.8	-4.6	-4.7	-5.1	-5.1	0.0	-2.4	-5.8	-5.8	-6.2	-6.3	-8.2	-8.8	-9.3	-9.3	-9.3	-9.3
SB/MD/CS	-7.3	-9.3	-12.5	-12.5	-13.0	-13.0	-6.2	-8.6	-12.1	-12.2	-12.5	-12.6	-7.2	-7.7	-7.8	-7.9	-7.8	-7.8
DB/SD/NS	-7.1	-	-	-	-	-	-5.8	-	-	-	-	-	-6.9	-	-	-	-	-
DB/SD/CS	-12.3	-	-	-	-	-	-10.8	-	-	-	-	-	-6.6	-	-	-	-	-
DB/MD/NS	-7.1	-9.5	-11.6	-11.7	-12.1	-12.2	-5.8	-8.4	-11.6	-11.7	-12.1	-12.1	-6.9	-7.0	-8.1	-8.0	-8.0	-8.0
DB/MD/CS	-12.3	-14.9	-17.2	-17.3	-17.9	-18.0	-10.8	-13.4	-16.5	-16.6	-17.1	-17.1	-6.6	-6.6	-7.4	-7.4	-7.2	-7.2

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 64 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 12$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-9.2	-	-	-	-	-
SB/SD/CS	-5.7	-	-	-	-	-	-6.0	-	-	-	-	-	-9.5	-	-	-	-	-
SB/MD/NS	0.0	-0.8	-2.5	-2.6	-2.8	-2.9	0.0	-1.7	-3.6	-3.7	-3.9	-3.9	-9.2	-9.9	-10.1	-10.2	-10.1	-10.1
SB/MD/CS	-5.7	-6.9	-8.6	-8.6	-8.9	-9.0	-6.0	-7.8	-9.8	-9.9	-10.0	-10.1	-9.5	-10.0	-10.4	-10.4	-10.2	-10.3
DB/SD/NS	-6.5	-	-	-	-	-	-4.8	-	-	-	-	-	-7.4	-	-	-	-	-
DB/SD/CS	-10.5	-	-	-	-	-	-9.7	-	-	-	-	-	-8.3	-	-	-	-	-
DB/MD/NS	-6.5	-7.6	-8.9	-9.2	-9.4	-9.6	-4.8	-6.5	-8.3	-8.4	-8.6	-8.6	-7.4	-8.0	-8.4	-8.2	-8.3	-8.1
DB/MD/CS	-10.5	-11.4	-13.0	-13.1	-13.6	-13.7	-9.7	-11.4	-13.1	-13.3	-13.4	-13.5	-8.3	-9.1	-9.2	-9.3	-8.9	-9.0

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 65 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 16$, $\gamma = 0.5$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-3.1	-	-	-	-	-
SB/SD/CS	-4.3	-	-	-	-	-	-5.1	-	-	-	-	-	-3.9	-	-	-	-	-
SB/MD/NS	0.0	-0.8	-2.3	-2.4	-2.6	-2.7	0.0	-1.1	-2.2	-2.4	-2.4	-2.5	-3.1	-3.4	-3.0	-3.1	-2.9	-3.0
SB/MD/CS	-4.3	-5.3	-6.6	-6.7	-6.9	-6.9	-5.1	-6.5	-7.5	-7.7	-7.6	-7.8	-3.9	-4.3	-4.0	-4.2	-3.9	-4.0
DB/SD/NS	-6.4	-	-	-	-	-	-5.5	-	-	-	-	-	-2.1	-	-	-	-	-
DB/SD/CS	-9.6	-	-	-	-	-	-9.9	-	-	-	-	-	-3.4	-	-	-	-	-
DB/MD/NS	-6.4	-7.4	-8.9	-9.0	-9.1	-9.2	-5.5	-6.7	-7.8	-8.1	-8.1	-8.3	-2.1	-2.3	-1.8	-2.1	-1.9	-2.1
DB/MD/CS	-9.6	-10.5	-12.2	-12.3	-12.4	-12.5	-9.9	-11.0	-12.3	-12.5	-12.6	-12.7	-3.4	-3.6	-3.2	-3.3	-3.2	-3.2

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 66 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 2$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.2	-	-	-	-	-
SB/SD/CS	-8.0	-	-	-	-	-	-6.6	-	-	-	-	-	-9.8	-	-	-	-	-
SB/MD/NS	0.0	-2.6	-7.3	-7.3	-7.7	-7.7	0.0	-3.6	-9.1	-9.2	-9.6	-9.6	-11.2	-12.2	-13.0	-13.0	-12.9	-12.9
SB/MD/CS	-8.0	-10.6	-15.7	-15.7	-16.2	-16.2	-6.6	-10.2	-15.9	-15.9	-16.3	-16.3	-9.8	-10.8	-11.4	-11.4	-11.3	-11.3
DB/SD/NS	-4.0	-	-	-	-	-	-4.1	-	-	-	-	-	-11.3	-	-	-	-	-
DB/SD/CS	-10.7	-	-	-	-	-	-9.8	-	-	-	-	-	-10.3	-	-	-	-	-
DB/MD/NS	-4.0	-6.9	-11.5	-11.5	-12.1	-12.1	-4.1	-7.8	-13.0	-13.1	-13.6	-13.7	-11.3	-12.1	-12.7	-12.8	-12.7	-12.8
DB/MD/CS	-10.7	-13.4	-18.4	-18.4	-18.9	-18.9	-9.8	-13.4	-18.9	-19.0	-19.5	-19.5	-10.3	-11.3	-11.9	-11.9	-11.9	-11.9

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 67 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 4$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-11.6	-	-	-	-	-
SB/SD/CS	-8.6	-	-	-	-	-	-6.5	-	-	-	-	-	-9.7	-	-	-	-	-
SB/MD/NS	0.0	-1.5	-4.0	-4.0	-4.4	-4.4	0.0	-2.4	-5.4	-5.4	-5.7	-5.7	-11.6	-12.5	-12.9	-12.9	-12.7	-12.8
SB/MD/CS	-8.6	-10.2	-13.2	-13.2	-13.6	-13.6	-6.5	-9.0	-11.9	-11.9	-12.2	-12.2	-9.7	-10.6	-10.5	-10.5	-10.3	-10.3
DB/SD/NS	-6.3	-	-	-	-	-	-4.2	-	-	-	-	-	-9.5	-	-	-	-	-
DB/SD/CS	-13.1	-	-	-	-	-	-9.5	-	-	-	-	-	-8.0	-	-	-	-	-
DB/MD/NS	-6.3	-8.2	-10.5	-10.5	-11.0	-11.0	-4.2	-6.9	-9.8	-9.8	-10.0	-10.0	-9.5	-10.2	-10.8	-10.8	-10.5	-10.4
DB/MD/CS	-13.1	-14.8	-17.5	-17.5	-18.1	-18.1	-9.5	-12.0	-14.8	-14.9	-15.1	-15.1	-8.0	-8.7	-8.7	-8.7	-8.4	-8.4

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 68 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 6$, $\gamma = 0.75$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-12.3	-	-	-	-	-
SB/SD/CS	-6.9	-	-	-	-	-	-6.7	-	-	-	-	-	-12.1	-	-	-	-	-
SB/MD/NS	0.0	-0.6	-2.0	-2.1	-2.4	-2.4	0.0	-1.4	-2.8	-2.9	-3.2	-3.2	-12.3	-13.0	-13.0	-13.0	-13.0	-13.0
SB/MD/CS	-6.9	-7.7	-9.2	-9.2	-9.5	-9.5	-6.7	-8.4	-9.6	-9.7	-9.9	-9.9	-12.1	-12.9	-12.7	-12.8	-12.7	-12.7
DB/SD/NS	-5.9	-	-	-	-	-	-5.4	-	-	-	-	-	-11.7	-	-	-	-	-
DB/SD/CS	-11.3	-	-	-	-	-	-11.5	-	-	-	-	-	-12.5	-	-	-	-	-
DB/MD/NS	-5.9	-6.5	-7.8	-7.8	-8.2	-8.3	-5.4	-6.7	-7.9	-8.1	-8.1	-8.3	-11.7	-12.4	-12.3	-12.5	-12.2	-12.3
DB/MD/CS	-11.3	-11.9	-13.4	-13.4	-13.7	-13.8	-11.5	-13.0	-14.0	-14.1	-14.1	-14.2	-12.5	-13.3	-12.9	-12.9	-12.7	-12.7

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 69 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 8$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-4.6	-	-	-	-	-
SB/SD/CS	-5.0	-	-	-	-	-	-5.8	-	-	-	-	-	-5.4	-	-	-	-	-
SB/MD/NS	0.0	-0.6	-1.6	-1.6	-1.8	-1.8	0.0	-1.0	-1.4	-1.6	-1.6	-1.7	-4.6	-5.0	-4.5	-4.6	-4.4	-4.5
SB/MD/CS	-5.0	-5.6	-6.5	-6.6	-6.7	-6.7	-5.8	-7.0	-7.2	-7.4	-7.3	-7.5	-5.4	-5.9	-5.3	-5.5	-5.2	-5.4
DB/SD/NS	-6.0	-	-	-	-	-	-6.0	-	-	-	-	-	-4.5	-	-	-	-	-
DB/SD/CS	-10.3	-	-	-	-	-	-11.1	-	-	-	-	-	-5.4	-	-	-	-	-
DB/MD/NS	-6.0	-6.8	-7.6	-7.7	-8.1	-8.2	-6.0	-7.0	-7.2	-7.5	-7.4	-7.7	-4.5	-4.7	-4.2	-4.3	-3.9	-4.1
DB/MD/CS	-10.3	-10.9	-11.4	-11.6	-11.7	-11.9	-11.1	-12.0	-12.3	-12.5	-12.5	-12.7	-5.4	-5.8	-5.5	-5.7	-5.5	-5.5

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 70 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 12$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-12.3	-	-	-	-	-
SB/SD/CS	-8.6	-	-	-	-	-	-7.6	-	-	-	-	-	-11.3	-	-	-	-	-
SB/MD/NS	0.0	-2.1	-6.3	-6.3	-7.0	-7.0	0.0	-2.9	-8.0	-8.0	-8.6	-8.6	-12.3	-13.1	-13.9	-13.9	-13.8	-13.8
SB/MD/CS	-8.6	-10.7	-15.2	-15.2	-15.9	-15.9	-7.6	-10.6	-15.6	-15.6	-16.2	-16.2	-11.3	-12.3	-12.6	-12.6	-12.5	-12.5
DB/SD/NS	-4.4	-	-	-	-	-	-3.6	-	-	-	-	-	-11.7	-	-	-	-	-
DB/SD/CS	-11.6	-	-	-	-	-	-10.5	-	-	-	-	-	-11.2	-	-	-	-	-
DB/MD/NS	-4.4	-6.6	-10.5	-10.5	-11.2	-11.2	-3.6	-6.9	-11.8	-11.8	-12.5	-12.5	-11.7	-12.6	-13.5	-13.6	-13.6	-13.6
DB/MD/CS	-11.6	-13.6	-18.0	-18.0	-18.7	-18.7	-10.5	-13.5	-18.6	-18.6	-19.3	-19.3	-11.2	-12.1	-12.9	-12.9	-12.9	-12.9

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 71 **Average policy and layout improvement potential with $\kappa = 4$, $\beta = 16$, $\gamma = 0.75$.**

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-13.9	-	-	-	-	-
SB/SD/CS	-9.7	-	-	-	-	-	-7.8	-	-	-	-	-	-12.2	-	-	-	-	-
SB/MD/NS	0.0	-0.8	-3.2	-3.2	-3.7	-3.7	0.0	-1.4	-4.0	-4.0	-4.4	-4.4	-13.9	-14.4	-14.6	-14.6	-14.5	-14.5
SB/MD/CS	-9.7	-10.7	-13.3	-13.3	-13.8	-13.8	-7.8	-9.3	-11.7	-11.7	-12.0	-12.0	-12.2	-12.7	-12.4	-12.4	-12.1	-12.1
DB/SD/NS	-5.7	-	-	-	-	-	-4.8	-	-	-	-	-	-13.0	-	-	-	-	-
DB/SD/CS	-13.5	-	-	-	-	-	-11.8	-	-	-	-	-	-12.3	-	-	-	-	-
DB/MD/NS	-5.7	-6.5	-9.1	-9.1	-9.7	-9.7	-4.8	-6.3	-8.7	-8.7	-9.1	-9.1	-13.0	-13.6	-13.4	-13.4	-13.2	-13.2
DB/MD/CS	-13.5	-14.3	-16.9	-16.9	-17.5	-17.5	-11.8	-13.2	-15.5	-15.6	-15.7	-15.8	-12.3	-13.0	-12.5	-12.5	-12.1	-12.1

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 72 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 2$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-15.8	-	-	-	-	-
SB/SD/CS	-8.2	-	-	-	-	-	-7.9	-	-	-	-	-	-15.5	-	-	-	-	-
SB/MD/NS	0.0	-0.4	-1.6	-1.6	-1.9	-1.9	0.0	-0.9	-1.9	-1.9	-2.1	-2.1	-15.8	-16.3	-16.0	-16.1	-16.0	-16.0
SB/MD/CS	-8.2	-8.8	-9.7	-9.7	-10.0	-10.0	-7.9	-8.9	-9.8	-9.8	-10.0	-10.0	-15.5	-15.9	-15.9	-15.9	-15.8	-15.8
DB/SD/NS	-5.2	-	-	-	-	-	-5.2	-	-	-	-	-	-15.8	-	-	-	-	-
DB/SD/CS	-12.0	-	-	-	-	-	-12.6	-	-	-	-	-	-16.4	-	-	-	-	-
DB/MD/NS	-5.2	-5.8	-6.9	-6.9	-7.5	-7.5	-5.2	-6.4	-7.2	-7.5	-7.4	-7.6	-15.8	-16.4	-16.2	-16.3	-15.7	-15.8
DB/MD/CS	-12.0	-12.8	-13.7	-13.8	-14.1	-14.1	-12.6	-13.7	-14.2	-14.4	-14.5	-14.6	-16.4	-16.7	-16.3	-16.3	-16.2	-16.2

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 73 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 4$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-8.5	-	-	-	-	-
SB/SD/CS	-6.4	-	-	-	-	-	-6.8	-	-	-	-	-	-8.8	-	-	-	-	-
SB/MD/NS	0.0	-0.5	-1.3	-1.3	-1.5	-1.5	0.0	-0.7	-0.9	-1.1	-1.0	-1.2	-8.5	-8.7	-8.1	-8.2	-8.0	-8.2
SB/MD/CS	-6.4	-7.1	-7.4	-7.5	-7.6	-7.6	-6.8	-7.6	-7.7	-7.9	-7.8	-8.0	-8.8	-9.0	-8.7	-8.9	-8.7	-8.8
DB/SD/NS	-6.3	-	-	-	-	-	-6.0	-	-	-	-	-	-8.2	-	-	-	-	-
DB/SD/CS	-12.2	-	-	-	-	-	-11.7	-	-	-	-	-	-8.0	-	-	-	-	-
DB/MD/NS	-6.3	-6.9	-7.9	-8.0	-8.3	-8.3	-6.0	-7.2	-7.1	-7.6	-7.3	-7.7	-8.2	-8.8	-7.7	-8.1	-7.4	-7.8
DB/MD/CS	-12.2	-12.8	-13.0	-13.3	-13.2	-13.4	-11.7	-12.7	-12.8	-13.1	-13.0	-13.2	-8.0	-8.3	-8.3	-8.2	-8.2	-8.2

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 74 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 6$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-14.0	-	-	-	-	-
SB/SD/CS	-9.8	-	-	-	-	-	-8.5	-	-	-	-	-	-12.7	-	-	-	-	-
SB/MD/NS	0.0	-1.4	-5.2	-5.2	-5.9	-5.9	0.0	-2.6	-6.9	-6.9	-7.5	-7.5	-14.0	-15.0	-15.5	-15.5	-15.4	-15.4
SB/MD/CS	-9.8	-11.2	-15.3	-15.3	-15.9	-15.9	-8.5	-11.1	-15.3	-15.3	-15.9	-15.9	-12.7	-13.8	-13.9	-13.9	-13.8	-13.8
DB/SD/NS	-3.6	-	-	-	-	-	-3.3	-	-	-	-	-	-13.7	-	-	-	-	-
DB/SD/CS	-12.4	-	-	-	-	-	-11.0	-	-	-	-	-	-12.4	-	-	-	-	-
DB/MD/NS	-3.6	-5.1	-8.9	-8.9	-9.6	-9.6	-3.3	-5.9	-10.5	-10.5	-11.0	-11.0	-13.7	-14.7	-15.4	-15.4	-15.3	-15.3
DB/MD/CS	-12.4	-14.0	-18.2	-18.2	-18.8	-18.8	-11.0	-13.5	-18.0	-18.0	-18.6	-18.6	-12.4	-13.4	-13.6	-13.6	-13.6	-13.6

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 75 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 8$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-14.9	-	-	-	-	-
SB/SD/CS	-9.8	-	-	-	-	-	-8.3	-	-	-	-	-	-13.6	-	-	-	-	-
SB/MD/NS	0.0	-1.2	-3.4	-3.4	-3.8	-3.8	0.0	-1.7	-4.1	-4.1	-4.3	-4.3	-14.9	-15.4	-15.5	-15.5	-15.4	-15.4
SB/MD/CS	-9.8	-11.2	-13.4	-13.4	-13.9	-13.9	-8.3	-10.1	-12.2	-12.2	-12.4	-12.4	-13.6	-13.9	-13.8	-13.8	-13.6	-13.6
DB/SD/NS	-4.9	-	-	-	-	-	-4.1	-	-	-	-	-	-14.1	-	-	-	-	-
DB/SD/CS	-13.2	-	-	-	-	-	-11.6	-	-	-	-	-	-13.4	-	-	-	-	-
DB/MD/NS	-4.9	-6.0	-8.4	-8.4	-8.8	-8.8	-4.1	-5.8	-8.2	-8.2	-8.5	-8.5	-14.1	-14.8	-14.6	-14.6	-14.6	-14.6
DB/MD/CS	-13.2	-14.6	-16.9	-16.9	-17.4	-17.4	-11.6	-13.4	-15.5	-15.5	-15.7	-15.7	-13.4	-13.9	-13.5	-13.5	-13.3	-13.3

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 76 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 12$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-18.6	-	-	-	-	-
SB/SD/CS	-9.4	-	-	-	-	-	-8.3	-	-	-	-	-	-17.6	-	-	-	-	-
SB/MD/NS	0.0	-0.3	-1.5	-1.5	-1.7	-1.7	0.0	-0.9	-1.9	-1.9	-2.0	-2.0	-18.6	-19.1	-18.9	-18.9	-18.9	-18.9
SB/MD/CS	-9.4	-10.0	-11.0	-11.0	-11.2	-11.2	-8.3	-9.2	-10.2	-10.2	-10.3	-10.3	-17.6	-17.9	-17.8	-17.8	-17.8	-17.8
DB/SD/NS	-6.3	-	-	-	-	-	-5.4	-	-	-	-	-	-17.8	-	-	-	-	-
DB/SD/CS	-14.4	-	-	-	-	-	-12.7	-	-	-	-	-	-17.0	-	-	-	-	-
DB/MD/NS	-6.3	-6.8	-8.4	-8.4	-8.6	-8.7	-5.4	-6.2	-7.3	-7.3	-7.6	-7.6	-17.8	-18.1	-17.6	-17.6	-17.6	-17.6
DB/MD/CS	-14.4	-14.9	-16.1	-16.1	-16.4	-16.4	-12.7	-13.7	-14.6	-14.6	-14.8	-14.8	-17.0	-17.4	-17.1	-17.1	-17.0	-17.0

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

Table 77 Average policy and layout improvement potential with $\kappa = 4$, $\beta = 16$, $\gamma = 1.0$.

	Three cross-aisles						Five cross-aisles						Three vs. five cross-aisles					
# DPs	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
SB/SD/NS	0.0	-	-	-	-	-	0.0	-	-	-	-	-	-10.6	-	-	-	-	-
SB/SD/CS	-7.3	-	-	-	-	-	-7.3	-	-	-	-	-	-10.6	-	-	-	-	-
SB/MD/NS	0.0	-0.5	-1.1	-1.1	-1.3	-1.3	0.0	-0.6	-0.8	-0.9	-0.9	-1.0	-10.6	-10.7	-10.4	-10.4	-10.3	-10.4
SB/MD/CS	-7.3	-7.8	-8.2	-8.2	-8.3	-8.3	-7.3	-8.1	-8.2	-8.3	-8.3	-8.4	-10.6	-10.9	-10.6	-10.7	-10.6	-10.7
DB/SD/NS	-6.8	-	-	-	-	-	-6.2	-	-	-	-	-	-9.9	-	-	-	-	-
DB/SD/CS	-13.6	-	-	-	-	-	-12.7	-	-	-	-	-	-9.7	-	-	-	-	-
DB/MD/NS	-6.8	-7.5	-8.4	-8.6	-8.8	-8.9	-6.2	-7.1	-7.1	-7.4	-7.4	-7.6	-9.9	-10.1	-9.3	-9.5	-9.1	-9.3
DB/MD/CS	-13.6	-14.1	-14.5	-14.6	-14.8	-14.8	-12.7	-13.5	-13.6	-13.8	-13.8	-14.0	-9.7	-10.0	-9.6	-9.8	-9.6	-9.7

The table shows for $\kappa = 4$ the percentage deviation in total picking time for settings with three and five cross-aisles, taking the setting with policy (SB/SD/NS) and one depot as reference value. Further it shows the deviation between the three and the five cross-aisle settings, taking the respective three cross-aisle settings as reference value. Abbreviations hold as follows: SB - static batching, DB - dynamic batching, SD - single drop-off point, MD - multiple drop-off points, NS - no cartless subtours, CS - cartless subtours, #DPs - number of drop-off points.

References

Asahiro Y, Kawahara K, Miyano E (2012) NP-hardness of the sorting buffer problem on the uniform metric. *Discrete Applied Mathematics* 160:1453–1464.

Boysen N, De Koster RBM, Weidinger F (2019) Warehousing in the e-commerce era: A survey. *European Journal of Operational Research* 277(2):396–411.

Cambazard H, Catusse N (2018) Fixed-parameter algorithms for rectilinear Steiner tree and rectilinear traveling salesman problem in the plane. *European Journal of Operational Research* 270(2):419–429.

Pansart L, Catusse N, Cambazard H (2018) Exact algorithms for the picking problem. *Computers & Operations Research.* 100:117–127.

Ratliff HD, Rosenthal AS (1983) Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research* 31(3):507–521.

Theys C, Bräysy O, Dullaert W, Raa B (2010) Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research* 200(3):755–763.

Weidinger F, Boysen N, Schneider M (2019) Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research* 274(2):501–515.