WORKING PAPER SERIES

Order Picking in Narrow-Aisle Warehouses: A Fast Approach to Minimize Waiting Times

Sandra Hahn/André Scholz

Working Paper No. 6/2017

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

FACULTY OF ECONOMICS AND MANAGEMENT

Impressum (§ 5 TMG)
Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
Sandra Hahn, André Scholz
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Order Picking in Narrow-Aisle Warehouses: A Fast Approach to Minimize Waiting Times

S. Hahn, A. Scholz

Abstract

Mail order companies like Zalando or Amazon reported a significant increase regarding the number of incoming customer orders in recent years. Customers are served from a central distribution center (warehouse) where requested items of the orders have to be retrieved (picked) from their storage locations. The picking process is performed by human operators (order pickers) who are employed on a large scale in order to enable a fast processing of the orders. However, due to limited space, aisles are often very narrow in warehouses, and order pickers cannot pass or overtake each other. Thus, an order picker may have to wait until another picker has performed his/her operations. The arising waiting times may significantly increase the processing times of the orders, implying that a large number of pickers does not guarantee for small processing times. Therefore, in this paper, the impact of several problem parameters on the amount of waiting time is investigated first and situations are identified where the consideration of waiting times is inevitable for an efficient organization of the picking process. In the second part of the paper, a solution approach, namely a truncated branch-and-bound algorithm, is proposed which aims for the minimization of the waiting times. By means of extensive numerical experiments, it is demonstrated that this approach provides high-quality solutions within a very small amount of computing time.

Keywords: Order Picking, Picker Routing, Picker Blocking

Corresponding author:

André Scholz

Postbox 4120, 39016 Magdeburg, Germany

Phone: +49 391 67 51841

Fax: +49 391 67 48223

Email: andre.scholz@ovgu.de

1 Introduction

Zalando, a large mail order company, recorded an increase of the number of customer orders by more than 400% in recent years, as the number of orders amounted to 11.0 million in 2011, while 55.3 million orders were received in 2015 (Statista, 2016). When placing an order, customers have the possibility to choose express deliveries, guaranteeing the requested items to be delivered at the next work day. Recently, even same-day deliveries have been tested in some regions. Thus, being able to process customer orders very fast becomes more important in order to ensure customer satisfaction. Before the items requested by the customers can be shipped to the customer locations, the orders have to be processed in the distribution center (warehouse), i.e. the items have to be retrieved from their storage locations. In most warehouses, this is done by human operators (order pickers) who perform tours through the warehouse. For processing a huge number of orders within a short amount of time, many order pickers are employed who work in the warehouse at the same time.

Besides a large number of orders, companies are confronted with an increasing number of different articles to be stored (Hirschberg, 2015). Due to limited space, warehouses often include narrow picking aisles in order to maximize space utilization (Gue et al., 2006). However, in narrow aisles, order pickers can neither pass nor overtake each other. When two pickers work in a narrow aisle at the same time, a picker may have to wait until the other picker has completed the work in this aisle. This can cause severe problems, as waiting times may arise on a large scale and the advantage of the employment of a large number of pickers diminishes. Although it is known that waiting times have a significant negative impact on the processing times, waiting times are rarely taken into account in the literature when guiding pickers through the warehouse.

The intention of this paper is twofold. A large variety of analytical and simulation models exists which estimate the impact of several problem parameters on the waiting times. However, almost all approaches rely on the assumption that all storage locations have to be visited regardless of the locations of requested items. In order to provide more realistic insights, we conduct extensive numerical experiments for the evaluation of the impact of the parameters. Combinations of parameters are identified where waiting times are significant and its consideration is inevitable for an efficient organization of the picking process. In the second part of the paper, a solution approach is provided which takes the waiting times into account. In fact, we propose a truncated branch-and-bound algorithm, where waiting instructions are given to order pickers. Such instructions include information about the points in time when a picker has to wait and when he/she continues the tour. By means of this approach, the benefit of using more sophisticated waiting instructions as well as the impact of the decisions regarding the selection of the picker who has to wait are investigated.

The remainder of the paper is organized as follows. A detailed description of the problem is given in the next section. Section 3 comprises a literature review. First, the results obtained by means of analytical and simulation models are reviewed. Second, solution approaches are presented which deal with guiding order pickers through the warehouse while taking waiting times into account. In Section 4, the impact of several parameters on the waiting times is investigated. Since waiting times are significant for several parameter combinations, a truncated branch-and-bound algorithm is proposed which aims for the minimization of the waiting times (Section 5). Section 6 is devoted to the evaluation of the performance of the algorithm. The paper concludes with a summary and an outlook on future research opportunities.

2 Problem description

In manual picker-to-parts order picking systems, order pickers walk or ride through the warehouse in order to retrieve requested items from their storage locations. The storage locations are typically arranged in such a way that they constitute a block layout (Roodbergen, 2001). A picking area following a block layout includes two types of aisles: picking aisles and cross aisles. Picking aisles are of identical length and width and are arranged parallel to each other. Furthermore, they have to be entered to retrieve items as the items are stored on pallets or racks located on one side or even both sides of the picking aisles. Cross aisles are arranged orthogonally to the picking aisles. They do not contain any requested items, but cross aisles are required for enabling the pickers to proceed from one picking aisle to another. Cross aisles divide the picking area into blocks and the picking aisles into subaisles (see Fig. 1).

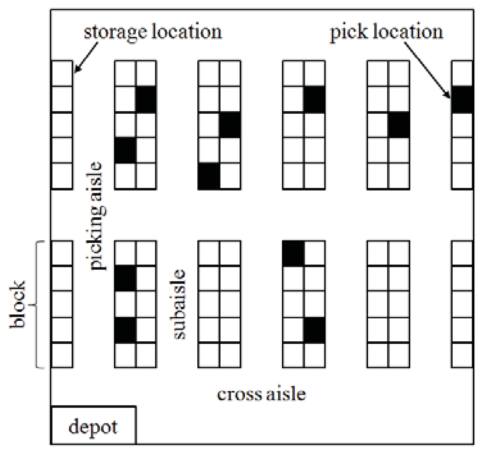


Fig. 1: Two-block layout

In Fig. 1, a two-block layout is depicted which contains 5 picking aisles and 10 subaisles. The rectangles symbolize the storage locations while the locations of requested items (pick locations) are represented by black rectangles. The depot is located in front of the leftmost picking aisle. A picker tour then starts at the depot, proceeds to the respective pick locations and ends at the depot. The time that an order picker needs for performing a tour (processing time) is composed of (Tompkins et al., 2010) the time required for preparing the tour (setup time), the time spent at the pick locations for the identification and the retrieval of the items (pick time) and the time needed for traveling from the depot to the pick locations, between the pick locations and back to the depot (travel time). Since a picking area with narrow subaisles is considered, an additional component, namely the waiting time, has to be taken into account. Waiting times arise because order pickers can neither pass nor overtake each other in narrow subaisles. Thus, several order pickers working in the same subaisle at the same time may cause congestion (blocking). (Note that congestion is not an issue in cross aisles.) A situation where an order picker cannot continue his/her operations because he/she is not able to pass or overtake another picker is referred to as a blocking situation. An example for a blocking situation is depicted in Fig. 2. Here, picker #1 is retrieving an item from its storage location. At the same time, picker #2 has to pass this location in order to reach another pick location. Due to the narrow subaisle, picker #2 is not able to pass picker #1. Thus, picker #2 has to wait until picker #1 has completed the retrieval of the item (assuming that picker #1 will proceed the tour by going upwards).

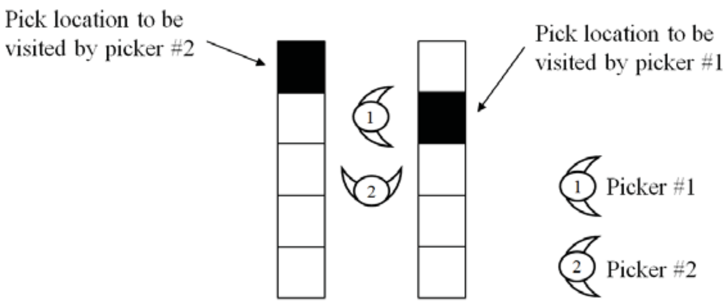


Fig. 2: Two order pickers working in the same subaisle

From the components of the processing time, the setup time and the pick time can be considered as constants (Bozer & Kile, 2008; Henn et al., 2010). The travel time is dependent on the sequence according to which the pick locations are meant to be visited. This sequence is determined by means of a certain procedure here, e.g. by application of a routing strategy (Roodbergen, 2001) or even by using an exact approach (Ratliff & Rosenthal, 1983; Roodbergen & de Koster, 2001). Thus, the sequence and, therefore, also the travel time can be assumed to be known for a given set of requested items, leaving the waiting time as the only variable component of the processing time. The waiting time of an order is dependent on the waiting instructions given to the picker performing the corresponding tour. A waiting instruction may have to be executed when a blocking situation arises. It comprises information about the point in time when a picker starts to wait (which also defines the position in the picking area where the picker waits) and the point in time when the picker continues the tour.

A set of customer orders to be processed is given. Each customer order is specified by the date when it has become available at the warehouse (arrival date) and by the requested articles and the respective quantities. The orders are processed by the pickers in the sequence they arrived at the warehouse (first-come-first-served) while a separate tour is performed for processing each customer order. As soon as an order picker becomes available, i.e. when he/she has finished a tour, he/she immediately starts with processing the next order in the sequence.

It is of prime importance to process customer orders as fast as possible. Therefore, the minimization of the throughput time of all orders (total throughput time) is a very common objective in this context (Le-Duc & de Koster, 2007; Van Nieuwenhuyse & de Koster, 2009; Yu & de Koster, 2009). The throughput time of an order is defined as the difference between the completion date of the order, i.e. the point in time when all requested items have been brought to the depot, and its arrival date. The throughput time of an order is composed of the time that elapsed after its arrival until processing of the order has started (start date) and its processing time. The start date of an order cannot be affected directly, as the sequence is given according to which the orders are processed. Indirectly, it is affected by the processing times of the orders processed before. Concerning the processing time of an order, as mentioned before, the waiting time is the only variable part. Thus, the minimization of the waiting times of all orders (total waiting time) is equivalent to the minimization of the total throughput time here.

The problem can now be stated as follows. Let a set of customer orders with known arrival dates be given including certain requested items. The customer orders are processed by a certain number of order pickers according to the sequence in which they arrived. Each picker processes the next order as soon as the picker becomes available. Furthermore, let setup times, pick times per item as well as a constant travel velocity of the pickers be given. In addition, the layout of the picking area is known and a routing algorithm for the construction of the picker tours is given. Then, for each order picker, the points in time when the picker has to wait and when he/she has to continue the tour have to be determined, respectively, in such a way that the total waiting time is minimized.

In the following section, the related literature is reviewed. First, we focus on analyses of the impact of several problem parameters on different performance criteria regarding the efficiency of the picking process. Second, solution approaches to related problems are reviewed.

3 Literature review

3.1 Analyses regarding the impact of parameters

A general consensus has been reached in the literature regarding the point that picker blocking may have a significant negative impact on the efficiency of the picking process. Thus, it is not surprising that a large variety of approaches exists concerning the estimation of the impact of certain problem parameters on performance criteria related to the picking process such as the total throughput time or the proportion of the total waiting time as part of the processing time of all orders (total processing time). All approaches deal with analytical and simulation models where most of them are based on the assumption that all subaisles are traversed according to a given sequence and direction (regardless of the fact whether a subaisle includes a pick location or not). It can then be assumed that the storage locations constitute a cycle. The order pickers start at a certain point of the cycle which represents the depot. From this point, they walk through the cycle until they reach this point again. With a probability of p (referred to as the pick density), an order picker stops at a storage location in order to retrieve an item.

Parikh & Meller (2009) dealt with picker blocking arising in warehouses with wide aisles, i.e. order pickers are able to pass and overtake each other in all aisles. However, pickers may block each other when the same pick location has to be visited at the same time (pick-face blocking). The authors pointed out that the proportion of the waiting time increases with an increasing pick density p. When p exceeds a certain value, waiting times decrease with a further increasing p. If p is equal to 1, no waiting times will arise as the pickers will stop at each location, implying that all pickers need the same time for performing a tour through the cycle. Furthermore, Parikh & Meller (2009) observed that a larger number of storage locations results in shorter waiting times, whereas the proportion of the total waiting time significantly increases when a larger number of pickers is available.

Skufca (2005) considered the impact of the number of order pickers, the number of storage locations and the pick density on the proportion of the total waiting time as part of the total processing time. The author dealt with a narrow-aisle warehouse, i.e. waiting times may arise since passing and overtaking of order pickers is not possible in subaisles (in-the-aisle blocking). Regarding the impact of the parameters mentioned above, Skufca (2005) obtained the same results as Parikh & Meller (2009). Based on the same assumptions, Gue et al. (2006) investigated the impact of the pick density but also of the pick-walk-time ratio, i.e. the average pick time per item divided by the time required for passing a storage location without retrieving an item. Gue et al. (2006) observed that an increasing pick-walk-time ratio leads to an increasing proportion of the total waiting time. Parikh & Meller (2010) additionally pointed out that waiting times may be underestimated by far if deterministic pick times per item are assumed. For example, no waiting times occur if the pick density equals 1 and pick times are deterministic. This is not true in case of non-deterministic pick times.

Pan & Shih (2008) and Pan & Wu (2012) are the only publications in which the picking area is not assumed to be cyclic as picker tours through the narrow-aisle warehouse are constructed by means of certain routing strategies. Pan & Shih (2008) applied the S-shape strategy. According to this strategy, each subaisle containing at least one requested item is traversed. An exception may occur in the last subaisle of a block where the picker returns after having retrieved all items in this aisle if this leads to a shorter tour. Pan & Shih (2008) investigated the impact of the procedure according to which articles are assigned to storage locations (storage assignment policy) on the throughput rate. The throughput rate is defined as the number of items retrieved within a certain amount of time. They compared a random storage assignment policy to a storage assignment policy of Jarvis & McDowell (1991) which is based on the demand frequency of the articles. Pan & Shih (2008) observed that application of the random assignment policy results in higher throughput rates. Pan & Wu (2012) chose the total throughput time as the performance criterion and extended the considerations of Pan & Shih (2008) to further routing strategies and several class-based storage assignment policies. They pointed out that the routing strategy leading to the shortest tours in combination with the across-aisle storage assignment policy (Petersen & Schmenner, 1999) results in the smallest total throughput time.

3.2 Solution approaches to related problems

Although the impact of waiting times on the performance of the picking process has widely been studied and observed to be significant in many cases, only few solution approaches exist which actually take waiting times into account when guiding order pickers through narrow-aisle warehouses. In fact, two approaches are available which address problems related to the one described in Section 2.

The scenario considered by Chen et al. (2013) differs from the problem defined in Section 2 regarding three aspects. First, the number of order pickers is restricted to two. Second, the next customer orders are not processed before both pickers have finished their tours. Third, no routing algorithm is given. Chen et al. (2013) proposed an ant colony optimization (ACO) approach to the resulting problem. By means of the ACO algorithm, a tour is constructed for the picker who leaves the depot first. This tour will remain unchanged. The ACO is then used for the determination of the tour of the other picker. The construction of the tour is based on the logical distance between pick locations, which is composed of the travel time between the locations as well as the waiting time caused by the tour of the first picker. Solving an instance with 2 customer orders containing up to 30 items requires 10 seconds of computing time. However, in terms of solution quality, the performance of this approach is hardly better than the performance of a modified S-shape strategy.

Chen et al. (2016) extended the considerations of Chen et al. (2013) to the case of an arbitrary number of pickers. They also designed an ACO approach to tackle this problem. First, the ACO algorithm is applied to construct the tours for all pickers without taking waiting times into account. Thus, as it is the case for the problem described in Section 2, a routing algorithm is given by which tours are determined beforehand. In a second step, blocking situations are identified. If a blocking situation is caused by two order pickers performing picking operations in the same subaisle, then the order picker who enters the subaisle first will perform the operations while the other picker waits at the entrance of this subaisle until he/she can execute the operations without being blocked. If two pickers block each other and at least one of the pickers traverses the subaisle without retrieving items, then it is checked whether the total throughput time can be decreased by traversing another subaisle, i.e. tours are allowed to be altered in the settings of Chen et al. (2016). The authors applied their approach to instances with 10 pickers and 30 requested items per order. Computing times have not been reported. The algorithm does not lead to convincing results concerning the solution quality as solutions provided by simple modifications of the S-shape and the largest gap strategy (Hall, 1993) cannot be improved significantly.

4 Evaluation of the impact of parameters on waiting times

4.1 Test instances

In the literature, several problem parameters have been identified which have an impact on the efficiency of the picking process in narrow-aisle warehouses (see Subsection 3.1). Since most approaches rely on the assumption that all subaisles are visited regardless of the pick locations, we conducted extensive numerical experiments in order to investigate the impact of the parameters on the performance of the picking process for more realistic settings. For the evaluation of the performance, the proportion of the total waiting time as part of the total processing time has been used as done by Skufca (2005), Gue et al. (2006) and Parikh & Meller (2009). Based on the observations from the literature, the impact of the following parameters is analyzed: the number of blocks, the number of picking aisles, the number of pickers, the pick-walk-time ratio, the number of items per order, the storage assignment policy, and the routing algorithm.

In the experiments, the picking area follows a block layout with $b \in \{1,2,3\}$ blocks and $m \in \{5,10\}$ picking aisles. Each subaisle contains 25 storage locations on each side, respectively. The distance between adjacent storage locations amounts to 1 length unit (LU). The same distance has to be covered for entering or leaving a subaisle. The distance between two adjacent picking aisles equals 5 LUs while 1.5 LUs are covered for traveling from the depot to the leftmost picking aisle (Henn & Wäscher, 2012).

Instances with 100 customer orders are considered. The number of requested items per order is uniformly distributed between n_l and n_u with $(n_l, n_u) \in \{(5,25),(10,50)\}$. For the assignment of articles to storage locations, two different procedures are applied, namely the random assignment policy ($a = r$) and the class-based assignment policy ($a = c$) used by Henn & Wäscher (2012). According to the random assignment policy, each storage location has the same probability of being a pick location. In the class-based assignment policy, articles are divided into three classes A, B and C based on the demand frequency. Class A articles are 10% of the articles with the highest demand and account for 52% of the total demand. 30% of all articles are assigned to class B where these articles represent 36% of the total demand. The remaining articles belong to class C and are characterized by quite low demand frequencies. Based on the class, articles are assigned to subaisles. Class A articles are located in 10% of the subaisles nearest to the depot while articles assigned to class C are situated in 60% of the subaisles farthest from the depot. The remaining subaisles include articles from class B. Each article is randomly assigned to a storage location of the corresponding subaisles.

For processing the customer orders, $k \in \{2,3,5,7\}$ order pickers are available. The time that an order picker needs to perform the tasks (see Section 2) is set as follows (Henn, 2015). The setup time amounts to 180 seconds while the picker needs 3 seconds to cover 1 LU. Since the pick-walk-time ratio α is usually 20 or less in practical applications (Gue et al., 2006), $\alpha \in \{3,10,20\}$ is chosen. This implies that a picker needs 9, 30 or 60 seconds for searching and retrieving an item. The picker tours are generated by means of two routing algorithms, namely the S-shape strategy and the Lin-Kernighan-Helsgaun (LKH) heuristic of Helsgaun (2000). The S-shape strategy represents the routing strategy most frequently used in practice (Roodbergen, 2001), while the LKH heuristic leads to very short tours (Theys et al., 2010).

The combination of all parameters mentioned above results in 576 problem classes. For each class, 48 instances have been generated, leading to 27648 instances in total.

4.2 Test solution approaches

For the determination of the proportion of the total waiting time as part of the total processing time, the problem described in Section 2 has to be solved. Two solution approaches are considered which are adapted from approaches proposed in the literature. In both approaches, based on the given picker tours, blocking situations are identified and waiting instructions are given. More precisely, the approaches work as follows. Let k denote the number of order pickers. In the first step, based on their arrival dates, the first k customer orders are processed by the pickers where each picker processes exactly one order. It is then checked whether blocking situations have to be dealt with. Blocking situations are identified chronologically, i.e. the situation which occurs first is considered. A blocking situation always concerns two order pickers. In order to deal with a blocking situation, waiting instructions are given to one of the pickers. The waiting instructions will not change the tour but they may affect the points in time when an order picker is at a certain location. Thus, these points in time have to be updated. Then, the next blocking situation is identified and dealt with based on the updated points in time. When all blocking situations have been considered which arise until one of the pickers has finished his/her current tour, the next customer order is assigned to this picker, and it is again checked whether new blocking situations arise. The procedure is repeated until all customer orders have been assigned to the pickers and all blocking situations have been dealt with. This principle is the same for both approaches presented below. However, the approaches differ with respect to the waiting instructions given to the pickers.

The first approach (A_1) is based on an approach of Ho & Chien (2006). They considered a distribution center in Taiwan where a single order picker was allowed to be in a subaisle only. Thus, an order picker is only permitted to enter a subaisle if no other picker is currently working in this subaisle. Otherwise, the picker has to wait at the entrance of the subaisle until the other picker has left the subaisle. Based on this rule, waiting instructions are given, i.e. the points in time when an order picker has to wait and when he/she has to proceed the tour are determined.

When applying A_1, waiting times can be expected to be very large, as pickers may have to wait although they would not actually block each other according to the definition of a blocking situation (see Section 2). Therefore, in the second approach (A_2), several order pickers are allowed to be in the same subaisle at the same time. In this approach, a blocking situation is dealt with by giving waiting instructions to the picker who left the depot at a later point in time (Chen et al., 2013). Thus, it is known which picker is allowed to continue the tour and which picker has to wait. Waiting instructions for a blocking situation can then be given in such a way that the waiting time caused by this situation is minimized. For the identification of such waiting instructions, the possible paths through subaisles are considered. A picker either traverses a subaisle or returns at a certain point. Moreover, either two pickers enter a subaisle from the same cross aisle or they use different cross aisles. Based on these observations, six blocking situations have to be considered (see Fig. 3).

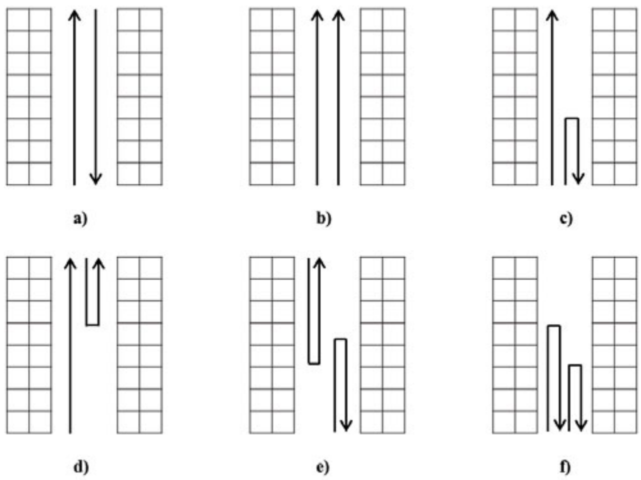


Fig. 3: Possible blocking situations in a subaisle

Waiting instructions are now given to the picker who has to wait based on the classification of the blocking situation. If the blocking situation follows a scenario depicted in Fig. 3 a) to e), the picker waits at the entrance of the subaisle until he/she can proceed the tour without blocking the other picker. The scenario shown in Fig. 3 f) is the only scenario where a picker may wait in the subaisle. This depends on the locations of the return points. If the location of the return point of the picker who has to wait is closer to the cross aisle from where the subaisle has been entered, then the picker will wait at the entrance as done in the other scenarios. If the return location is farther away and if the picker has entered the subaisle first, then he/she may wait at the return location until he/she can proceed the tour without being blocked by the other picker.

4.3 Results

The results of the experiments are depicted in Tables 1 and 2 and Tables A1 to A6, where Tables denoted by an "A" are included in the appendix available at `http://www.mansci.ovgu.de/mansci/en/Research/Materials/2017+_+I_-p-632.html`. Tables 1, 2, A1 and A2 contain information about the average proportion of the total waiting time as part of the total processing time (in %) for the approaches A_1 and A_2 combined with the S-shape strategy and the LKH heuristic, respectively. The corresponding total processing times (in hours) are shown in Tables A3 to A6.

Concerning the routing algorithms, the results are very similar. Both algorithms result in the smallest

Table 1: Proportion [%] of the total waiting time as part of the total processing time for A_1 and the S-shape strategy

α	(n_l, n_u)	b	m	$k=2$		$k=3$		$k=5$		$k=7$	
				$a=r$	$a=c$	$a=r$	$a=c$	$a=r$	$a=c$	$a=r$	$a=c$
3	(5,25)	1	5	1.6	2.3	3.5	5.2	8.0	13.0	15.4	28.6
3	(5,25)	1	10	1.1	1.9	2.2	3.9	4.6	9.5	7.4	18.4
3	(5,25)	2	5	1.4	2.5	3.1	5.3	6.4	13.0	10.1	24.4
3	(5,25)	2	10	0.9	1.3	2.0	2.9	4.0	6.9	6.1	11.9
3	(5,25)	3	5	1.0	1.8	2.3	3.6	4.8	8.2	7.8	13.1
3	(5,25)	3	10	0.6	1.1	1.2	2.2	2.7	5.0	4.4	8.4
3	(10,50)	1	5	2.0	2.9	4.3	6.3	9.8	17.8	20.5	37.1
3	(10,50)	1	10	0.9	2.0	1.9	4.6	4.2	10.9	6.6	23.3
3	(10,50)	2	5	1.4	2.9	2.9	6.0	6.0	14.5	9.3	29.1
3	(10,50)	2	10	0.8	1.3	1.7	2.7	3.5	6.4	5.6	10.8
3	(10,50)	3	5	1.0	1.6	2.1	3.4	4.3	7.6	6.9	12.4
3	(10,50)	3	10	0.6	1.0	1.1	2.0	2.5	4.4	4.3	7.4
10	(5,25)	1	5	2.7	4.4	6.0	10.1	13.3	29.0	23.1	47.6
10	(5,25)	1	10	1.4	3.5	3.1	7.8	6.3	21.3	10.2	40.5
10	(5,25)	2	5	1.8	4.1	3.8	9.6	8.1	26.0	12.6	44.3
10	(5,25)	2	10	1.2	2.1	2.1	4.7	4.4	10.7	7.1	18.5
10	(5,25)	3	5	1.4	2.4	2.8	5.3	5.9	12.1	9.2	20.2
10	(5,25)	3	10	0.7	1.5	1.4	3.2	3.0	7.1	4.9	11.6
10	(10,50)	1	5	4.1	6.0	8.2	14.0	16.7	38.4	27.3	55.3
10	(10,50)	1	10	1.7	4.5	3.6	10.1	7.7	29.3	11.7	48.5
10	(10,50)	2	5	2.2	5.5	4.5	11.8	9.4	33.2	14.0	51.0
10	(10,50)	2	10	1.0	2.3	2.1	4.9	4.6	11.7	7.1	19.6
10	(10,50)	3	5	1.3	2.9	2.9	5.7	6.3	12.7	9.4	21.1
10	(10,50)	3	10	0.7	1.6	1.5	3.2	3.3	7.1	4.9	11.7
20	(5,25)	1	5	4.1	6.8	8.2	16.6	18.0	42.0	28.7	57.6
20	(5,25)	1	10	1.9	5.4	3.9	12.7	8.2	34.8	13.2	52.6
20	(5,25)	2	5	2.5	6.4	4.9	14.6	9.9	37.3	15.4	54.6
20	(5,25)	2	10	1.2	3.1	2.4	6.4	5.1	14.5	8.3	25.0
20	(5,25)	3	5	1.6	3.4	3.3	7.4	7.2	16.2	10.9	26.8
20	(5,25)	3	10	0.8	2.0	1.7	4.1	3.6	9.6	5.6	15.5
20	(10,50)	1	5	5.8	8.8	10.8	20.2	21.0	47.0	32.1	61.7
20	(10,50)	1	10	2.6	6.7	5.4	16.2	10.5	42.1	15.6	58.1
20	(10,50)	2	5	2.9	7.5	5.5	18.0	11.5	43.7	17.5	59.2
20	(10,50)	2	10	1.4	3.2	2.6	7.0	5.6	16.3	8.4	28.0
20	(10,50)	3	5	1.9	3.8	3.8	7.9	8.0	17.0	11.7	28.4
20	(10,50)	3	10	0.9	2.3	1.8	4.6	3.8	9.5	5.9	16.0

proportion of the total waiting time for the problem class ($\alpha = 3$, $(n_l, n_u) = (5, 25)$, $b = 3$, $m = 10$, $k = 2$, $a = r$). When applying A_1 the smallest proportion of the waiting time amounts to 0.6% for both routing algorithms, while the proportion equals 0.4% and 0.5% for the S-shape strategy and the LKH heuristic when A_2 is used. For the S-shape strategy, the maximum proportion of the waiting time is 61.7% for A_1 and 22.2% for A_2. Regarding the LKH heuristic, proportions of up to 62.2% and 28.6% can be observed. On average, order pickers wait for 10.9% or 4.7% of the total processing time if A_1 or A_2 is applied and tours are constructed by means of the S-shape strategy. When using the LKH heuristic, waiting times account for 12.3% or 6.0% of the total processing time. It can be seen that the proportions are slightly

Table 2: Proportion [%] of the total waiting time as part of the total processing time for A_2 and the S-shape strategy

α	(n_l, n_u)	b	m	$k=2$		$k=3$		$k=5$		$k=7$	
				$a=r$	$a=c$	$a=r$	$a=c$	$a=r$	$a=c$	$a=r$	$a=c$
3	(5,25)	1	5	1.3	0.9	2.9	2.0	6.1	3.9	10.0	6.1
3	(5,25)	1	10	1.0	0.7	1.8	1.3	3.7	2.8	5.7	4.5
3	(5,25)	2	5	1.1	1.0	2.1	2.0	4.6	4.3	6.7	7.0
3	(5,25)	2	10	0.6	0.9	1.4	2.0	2.7	4.1	4.3	6.4
3	(5,25)	3	5	0.7	0.8	1.5	1.6	3.2	3.7	4.9	5.6
3	(5,25)	3	10	0.4	0.6	0.8	1.3	1.7	2.8	2.8	4.3
3	(10,50)	1	5	1.7	1.1	3.6	2.4	7.7	4.6	13.8	6.9
3	(10,50)	1	10	0.7	0.6	1.5	1.3	3.1	2.7	4.7	4.2
3	(10,50)	2	5	1.2	1.4	2.4	2.7	5.0	5.7	7.3	8.8
3	(10,50)	2	10	0.7	1.0	1.3	2.0	2.8	4.4	4.4	6.8
3	(10,50)	3	5	0.8	0.9	1.8	1.9	3.4	3.9	5.5	6.1
3	(10,50)	3	10	0.5	0.6	0.9	1.1	1.9	2.4	3.1	3.7
10	(5,25)	1	5	2.3	1.7	4.5	3.4	9.6	6.6	15.0	10.4
10	(5,25)	1	10	1.2	1.2	2.7	2.4	5.3	4.9	8.0	7.8
10	(5,25)	2	5	1.4	1.4	2.8	3.2	6.0	6.4	8.9	10.0
10	(5,25)	2	10	0.9	1.5	1.6	2.9	3.5	6.1	5.5	9.5
10	(5,25)	3	5	1.0	1.2	2.1	2.4	4.2	5.3	6.5	8.3
10	(5,25)	3	10	0.5	1.0	1.0	1.9	2.1	4.1	3.4	6.2
10	(10,50)	1	5	3.1	2.2	6.2	4.2	12.2	8.2	18.7	11.9
10	(10,50)	1	10	1.4	1.3	2.8	2.7	5.8	5.2	8.4	7.9
10	(10,50)	2	5	1.8	2.1	3.4	4.0	7.1	8.0	10.7	12.1
10	(10,50)	2	10	0.9	1.7	1.9	3.5	3.9	7.1	5.9	11.3
10	(10,50)	3	5	1.1	1.4	2.5	2.6	5.1	5.3	7.7	8.2
10	(10,50)	3	10	0.6	0.8	1.3	1.7	2.7	3.8	4.0	5.7
20	(5,25)	1	5	3.0	2.7	6.2	5.1	12.5	9.8	18.8	14.3
20	(5,25)	1	10	1.8	1.8	3.4	3.5	7.0	7.2	10.4	11.3
20	(5,25)	2	5	1.9	2.2	3.7	4.5	7.4	8.8	10.9	13.7
20	(5,25)	2	10	1.1	1.9	2.0	3.9	4.2	8.4	6.6	13.1
20	(5,25)	3	5	1.3	1.8	2.6	3.4	5.3	7.2	7.8	10.9
20	(5,25)	3	10	0.7	1.3	1.3	2.6	2.7	5.9	4.3	8.7
20	(10,50)	1	5	4.3	3.0	7.7	6.2	14.8	10.7	22.2	14.9
20	(10,50)	1	10	2.1	2.0	4.2	3.8	7.9	7.6	11.6	11.5
20	(10,50)	2	5	2.4	2.6	4.6	5.4	9.2	10.2	13.5	15.0
20	(10,50)	2	10	1.3	2.0	2.5	4.4	4.9	9.4	7.3	14.5
20	(10,50)	3	5	1.6	1.7	3.1	3.4	6.3	6.8	9.1	10.5
20	(10,50)	3	10	0.7	1.2	1.6	2.4	3.3	5.0	5.2	7.7

larger if tours are generated by application of the LKH heuristic. This can be explained by the fact that the LKH heuristic constructs shorter tours, resulting in smaller total processing times (see Tables A3 to A6). The impact of the other parameters is nearly the same for both routing algorithms. Therefore, the analysis is based on the results related to the S-shape strategy only.

Number of order pickers

According to the literature, waiting times significantly increase with a rising number of order pickers k since more blocking situations arise when many pickers work in the same picking area at the same time.

This is also verified by the results of the experiments. While the average proportion of the total waiting time amounts to 2.5% and 1.4% for A_1 and A_2 when the number of order pickers is very small ($k = 2$), the pickers spend 21.1% and 8.7% of their time on waiting for other pickers performing their operations, respectively, if many pickers are simultaneously employed ($k = 7$).

Size of the picking area and storage assignment policy

The size of the picking area is dependent on the number of blocks b and the number of picking aisles m. If the picking area is quite large, order pickers do not come across each other very often, resulting in few blocking situations and a short total waiting time. The same line of argumentation holds for the application of the random assignment policy instead of using the class-based storage assignment strategy. Thus, it can be expected that the proportion of the total waiting time decreases with increasing values for b and m, and that the proportion is smaller for the random assignment policy. In Table 3, the average proportion of the total waiting time is depicted for A_1 and A_2 dependent on the number of blocks, the number of picking aisles and the storage assignment policy.

Table 3: Proportion [%] of the total waiting time dependent on the size of the warehouse and the storage assignment policy

b	m	A_1		A_2	
		$a = r$	$a = c$	$a = r$	$a = c$
1	5	12.3	24.1	8.7	6.0
1	10	5.7	19.5	4.4	4.2
2	5	7.0	21.9	5.3	5.9
2	10	3.7	9.3	3.0	5.4
3	5	4.9	10.2	3.7	4.4
3	10	2.6	5.9	2.0	3.2

As can be seen in Table 3, the results of the experiments match with the expectations if A_1 is applied. The proportion of the total waiting time decreases with an increasing number of blocks (15.4% for $b = 1$, 10.4% for $b = 2$ and 5.9% for $b = 3$), it decreases with a rising number of picking aisles (13.4% for $m = 5$ and 7.8% for $m = 10$) and the proportion gets smaller when the random assignment policy is applied (15.1% for $a = c$ and 6.0% for $a = r$).

Concerning A_2, the impact of the size of the picking area is dependent on the storage assignment policy. For the random assignment policy ($a = r$), the proportions of the total waiting time are in line with the expectations as they decrease with increasing numbers of blocks (5.8% for $b = 1$, 4.9% for $b = 2$ and 3.3% for $b = 3$) and picking aisles (5.7% for $m = 5$ and 3.7% for $m = 10$). However, if the class-based assignment procedure is used ($a = c$) and the picking area contains 10 picking aisles, the proportions increase when switching from 1 block to 2 blocks. Furthermore, the proportion of the waiting time is larger for ($b = 3$, $m = 5$) than for ($b = 1$, $m = 10$), although the latter picking area contains fewer subaisles. This can be explained by the procedure how the articles are assigned to the three classes A, B and C. The articles included in classes A and B account for 88% of the total demand and they are distributed over the 40% of the subaisles which are nearest to the depot. In case of a two-block layout with 5 picking aisles, classes A and B are solely assigned to subaisles of the block nearest to the depot (first block). Thus, it is quite likely that all pick locations included in a tour are situated in the first block. Moreover, most of the subaisles assigned to classes A and B will have to be visited. Tours constructed by means of the S-shape strategy are then very similar as all of these subaisles are traversed. This results in quite short waiting times since order pickers may only be blocked by other pickers who currently retrieve an item. In contrast, if picking areas include 2 blocks and 10 picking aisles or 3 blocks, then subaisles of the second block are assigned to class B as well. Therefore, at least two blocks are part of the tours, which makes the resulting tours much more diverse. Order pickers then traverse subaisles in different directions. If an order picker is blocked by another picker who traverses the subaisle in a different direction, the picker has to wait until the other picker has left the subaisle. In most cases, this causes considerably larger waiting times than blocking situations where both pickers traverse an aisle in the same direction. Thus, the proportion of the total waiting time increases if not all frequently requested articles are assigned to subaisles of the first block.

Pick-walk-time ratio

In the experiments, the travel velocity of the pickers has been fixed and the time required for performing the operations at a pick location is varied. The larger the pick-walk-time ratio α gets, the longer an order picker stops at a pick location. Thus, it can be expected that a larger value for α leads to an increasing proportion of the total waiting time (Gue et al., 2006). The results of the experiments match with the expectations. For the application of A_1, the average proportion amounts to 6.5% for $\alpha = 3$, to 11.0% for $\alpha = 10$ and to 14.3% for $\alpha = 20$ while the average proportions equal 3.1%, 4.7% and 6.2%, respectively, if A_2 is used.

Number of requested items per customer order

According to Gue et al. (2006), waiting times increase with an increasing number of requested items per order. Furthermore, Hong et al. (2010) pointed out that a larger variance in the number of items will increase the proportion of the waiting time. Thus, it is expected that larger proportions can be observed for classes with $(n_l, n_u) = (10, 50)$. This is true for both A_1 and A_2 as the average proportions rise from 9.9% and 4.4% to 11.3% and 4.9%, respectively, when (n_l, n_u) are raised from $(5, 25)$ to $(10, 50)$.

Besides the impact of different problem parameters on the proportion of the total waiting time, the results of the experiments clearly show that, in many settings, taking waiting times into account is pivotal for achieving small processing times. In some settings, more than half of the total processing time can be attributed to waiting times, i.e. order pickers spend more time on waiting than on traveling through the warehouse and retrieving items. In order to keep waiting times at a reasonable level, a truncated branch-and-bound algorithm is presented in the next section.

5 A truncated branch-and-bound algorithm

5.1 General overview

In both solution approaches presented in Subsection 4.2, waiting instructions are given to a predefined picker (e.g. the picker who enters a subaisle at a later point in time). This may result in high waiting times. For example, if a picker enters a subaisle first but has to pick many items in this subaisle, then a picker, whose current tour only includes few pick locations in this subaisle, may have to wait for a long time. Therefore, a solution approach is presented which deals with the determination of the picker to whom waiting instructions are given. Since exactly two decisions are possible in this case, it seems reasonable to apply a branch-and-bound algorithm. Due to computing time and memory issues, a truncated branch-and-bound (TBB) algorithm has been designed. In TBB algorithms, the branching scheme of a branch-and-bound algorithm is kept while heuristic evaluation methods are applied to prune some branches (Rakrouki et al., 2012). By the heuristic pruning of branches, the computational effort is considerably reduced. However, optimality of the solution obtained cannot be guaranteed.

A pseudo-code of the TBB algorithm designed here is depicted below. In the TBB algorithm, each node of the tree represents a partial solution. At the beginning of the algorithm, no assignments of orders to pickers have been performed. The root r is then constructed by application of the expansion procedure. In the expansion procedure, (some) customer orders are assigned to order pickers and blocking situations are identified which arise by processing the orders according to the tours constructed by the given routing algorithm. After the expansion procedure is completed, the root is either assigned to the set of active nodes V or to the set of terminal solutions F. A node represents a terminal solution if all customer orders have been assigned to order pickers and all blocking situations have been taken into account. If r already corresponds to a terminal solution, the TBB algorithm terminates. Otherwise, iterations are performed as long as active nodes exist. An iteration starts with the selection of an active node $\tilde{v}$. A branching procedure is then applied to $\tilde{v}$, resulting in two nodes v_1 and v_2 located on the next level of the tree. By branching, the decision about the picker who has to wait is taken for a certain blocking situation and waiting instructions are given to this picker. The expansion procedure is applied to v_1 and v_2, respectively. It is then checked whether the nodes represent a terminal solution or have to be included in the set of active nodes. At the end of an iteration, the pruning procedure identifies active nodes which are excluded from the solution process, i.e. they are removed from the set of active nodes. At the end of the solution process, the TBB algorithm returns the node v^* which corresponds to the terminal solution resulting in the minimum total waiting time w. In the following, the components of the TBB algorithm are explained in greater detail.

Algorithm 1 Truncated Branch-and-Bound Algorithm

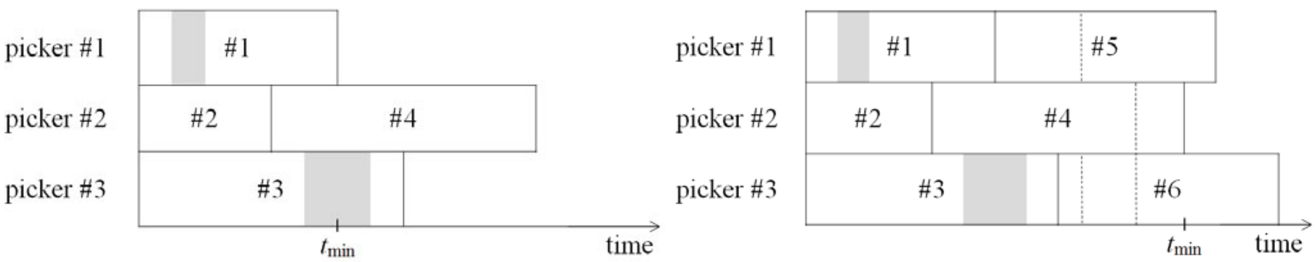
Input: problem data, node r corresponding to a partial solution with no orders assigned to pickers;

Output: node v^* corresponding to a solution to the problem defined in Section 2;

```
r := Expansion_Procedure(r);
if r corresponds to a terminal solution then
    V := ∅; F := {r};
else
    V := {r}; F := ∅;
end if
while V ≠ ∅ do
    ṽ := Node_Selection(V); V := V \ {ṽ};
    (v1, v2) := Branching_Procedure(ṽ);
    for v ∈ {v1, v2} do
        v := Expansion_Procedure(v);
        if v corresponds to a terminal solution then
            F := F ∪ {v};
        else
            V := V ∪ {v};
        end if
    end for
    for v ∈ V do
        Pruning_Procedure(v);
    end for
end while
v* := arg min {w(v) | v ∈ F};
```

5.2 Expansion of a node

As mentioned before, the TBB algorithm starts with no customer orders being assigned to the order pickers and it then successively assigns the orders to the pickers. This is done in the expansion procedure which works as follows. Let a node v be given, representing a partial solution with a set of customer orders already assigned to order pickers and a set of waiting instructions already given to the pickers. Let $t_{\min}$ denote the point in time when the next order picker becomes available where the calculation of $t_{\min}$ includes the waiting time of instructions already given but not waiting times which may arise due to blocking situations not yet taken into account. The node v can then be expanded if and only if all blocking situations arising until $t_{\min}$ have been dealt with in the corresponding partial solution. In this case, the next order in the sequence (based on the arrival date) is assigned to the picker who finishes the tour at $t_{\min}$ and $t_{\min}$ is updated. This procedure is repeated until at least one blocking situation arises. At the end of the expansion or if the node cannot be expanded, the expansion procedure returns the blocking situations which have been identified. An example of the expansion of a node is depicted in Fig. 4.



a) Partial solution before expansion of the node

b) Partial solution after expansion of the node

Fig. 4: Example of an expansion of a node

In Fig. 4a), a partial solution is given which corresponds to an expandable node. A Gantt chart is depicted where the rectangles represent the tours to be performed for processing the respective orders. The width of a rectangle gives information about the duration of a tour. The gray parts of a rectangle stand for the waiting time caused by executing the waiting instructions. It can be seen that customer orders #1 to #4 have already been assigned to order pickers in this partial solution. Furthermore, picker #1 and picker #3 execute a waiting instruction, respectively. Here, $t_{\min}$ is defined as the point in time when picker #1 has completed the tour. Thus, when expanding the node, the next order is assigned to picker #1 which is shown in Fig. 4b). Since no blocking situation results from this assignment, order #6 is assigned to picker #3 because this picker will be the next picker who is available. This assignment causes two blocking situations which are illustrated by the dotted lines in Fig. 4b). The first blocking situation concerns pickers #1 and #3 while the other situation relates to pickers #2 and #3. Thus, the expansion procedure terminates and $t_{\min}$ is now the point in time when picker #2 has processed customer order #4. (Note that the effect of the two blocking situations is not included in $t_{\min}$.)

5.3 Characteristics of a (partial) solution

Let $t_{\min}$ be defined as in the previous subsection. A (partial) solution is then characterized by:

1) the customer orders already assigned to a picker and the assignment of the orders to the pickers;

2) for each picker, the waiting instructions received for the blocking situations already dealt with;

3) the total waiting time caused by performing the received waiting instructions;

4) the number of assigned customer orders;

5) the number of remaining blocking situations arising until $t_{\min}$.

The characteristics mentioned in 1) and 2) contain information about the decisions to be taken for solving the problem described in Section 2. The third component represents the objective function value if the solution is a terminal solution. Otherwise, it defines a lower bound regarding the objective function value. Since the waiting time is dependent on the customer orders and the blocking situations already taken into account, components 4) and 5) are required for the identification of the pairs of the corresponding nodes which can be compared regarding the lower bounds in the pruning procedure. The selection of the node to be considered in an iteration is also based on these components.

5.4 Selection of a node, and branching and pruning procedures

In order to keep the tree at a reasonable size, the node to be considered is chosen in such a way that many nodes can be compared in the pruning procedure, i.e. the corresponding partial solutions of the nodes are equal with respect to the number of assigned customer orders. Therefore, a node is selected according to the following priorities:

1) the smallest number of assigned customer orders;

2) the smallest number of remaining blocking situations;

3) the smallest total waiting time;

4) the first generated node.

The branching procedure is then applied to the selected node. In this procedure, the first arising blocking situation identified in the expansion procedure is considered and two nodes are generated. The generation of the nodes is based on the decision regarding the picker who has to wait in this blocking situation. Waiting instructions are then given to the respective picker. The waiting instructions match with the instructions used in approach A_2 (see Subsection 4.2).

In a TBB algorithm, the pruning procedure replaces the bounding phase of a classic branch-and-bound algorithm. Here, two heuristic methods are applied to prune branches, reducing the size of the tree. The first procedure is based on the comparison of nodes regarding the lower bounds. As mentioned before, nodes can only be compared if they relate to partial solutions characterized by the same number of assigned customer orders. A branch corresponding to a node is then pruned if another node exists whose corresponding partial solution either contains fewer remaining blocking situations while not having a larger total waiting time or if the number of remaining blocking situations is equal for both partial solutions but the total waiting time is smaller for the other one.

The second possibility for pruning a branch of a node is related to the number of remaining blocking situations after application of the branching procedure. By branching, a blocking situation is taken into account and waiting instructions are given to a picker. In general, the number of remaining blocking situations either decreases by 1 or further orders can even be assigned until new blocking situations are identified. However, since the execution of waiting instructions results in changes in the points in time when the respective picker is at certain locations, it is also possible that blocking situations arise which did not occur before. Thus, the number of remaining blocking situations may remain unchanged or even increase. In this case, the branch of the generated node is pruned. An exception occurs if both resulting branches would have been pruned. The node whose corresponding partial solution shows the smaller number of remaining blocking situations is then kept, guaranteeing the algorithm to find a terminal solution.

6 Performance of the TBB algorithm

6.1 Setup

For the evaluation of the performance of the TBB algorithm, numerical experiments are conducted. The settings of the experiments are chosen according to the setup of the experiments described in Subsection 4.1. Based on the observations from Subsection 4.3, we focus on problem classes in which the proportion of the total waiting time as part of the total processing time can be expected to be significant. Thus, the problem classes with the following parameters are considered. For processing $N \in \{100, 200\}$ customer orders, each including between 10 and 50 items, $k \in \{3, 5, 7, 10\}$ order pickers are available while the pick-walk-time ratio α either amounts to 10 or is equal to 20. The picking area of the warehouse contains $b \in \{1,2\}$ blocks and $m \in \{5,10\}$ picking aisles. Articles are assigned to storage locations according to the random or the class-based assignment policy.

The combination of all parameter values gives rise to 256 problem classes. For each class, 48 problem instances have been generated, resulting in 12288 instances in total. The TBB algorithm has been implemented using Visual Studio C++ 2015. The numerical experiments have been executed by means of a Haswell system with up to 3.2 GHz and 16 GB RAM per core.

The performance of the TBB algorithm is evaluated with respect to the amount of improvement (in terms of the reduction of the total waiting time) obtained compared to the application of the approaches A_1 and A_2, i.e. the impact of the decisions regarding the given waiting instructions and the selection of the picker who has to wait is considered. Furthermore, computing times are reported in order to investigate whether the TBB algorithm is able to deal with large-sized instances.

6.2 Improvements obtained by application of the TBB algorithm

6.2.1 Improvements by allowing several pickers to work in the same aisle at the same time

In Tables A7 and A8, the results of the experiments are depicted for problem classes with 100 customer orders where tours have been constructed by means of the S-shape strategy and the LKH heuristic, respectively. Tables 4 and 5 include the respective results for problem classes containing 200 orders. For each problem class, the average total waiting times w_1, w_2 and w_B (in hours) are given which result by the application of the approaches A_1 and A_2 and the TBB algorithm. Furthermore, the average relative amount of reduction of the total waiting time imp_i (in %) is depicted which is obtained by applying the TBB algorithm instead of using approach A_i ($i \in \{1,2\}$).

According to approach A_1, only a single picker is allowed to be in a subaisle (see Subsection 4.2). If a picker has to enter a subaisle currently occupied by another picker, the picker has to wait until the other picker has performed the operations and has left this subaisle. In contrast, more sophisticated waiting instructions are given in the TBB algorithm. Thus, by comparing the total waiting times resulting by application of A_1 and the TBB algorithm, the impact of the waiting instructions on the waiting times can be analyzed. (Note that a further difference between A_1 and the TBB algorithm consists in the selection of the picker who has to wait. However, the impact on the solution quality is quite small compared to the impact of the waiting instructions.)

Table 4: Evaluation of the truncated brach-and-bound algorithm for problem classes with 200 customer orders and S-shape strategy

a	α	b	m	$k=3$					$k=5$					$k=7$					$k=10$				
				w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2
r	10	1	5	8.1	5.8	4.0	50.8	31.5	17.6	12.2	10.7	39.5	12.4	33.1	20.2	16.8	49.2	17.0	70.3	38.2	22.2	68.4	41.9
r	10	1	10	4.3	3.3	1.3	68.7	60.1	8.8	6.3	2.9	66.6	53.4	14.5	10.0	5.1	64.6	48.5	24.1	15.2	8.4	65.1	44.6
r	10	2	5	5.0	4.0	2.2	55.0	44.5	10.4	8.0	5.3	49.1	33.7	16.6	12.4	8.6	47.8	30.2	28.0	19.4	14.3	48.8	26.0
r	10	2	10	3.0	2.7	1.2	60.7	56.0	6.5	5.7	2.7	58.1	51.5	10.3	8.6	4.4	57.0	48.4	16.5	13.2	7.2	56.2	45.4
r	20	1	5	17.3	11.8	7.4	57.2	37.3	36.5	23.9	19.4	46.7	18.7	64.8	38.9	30.5	52.9	21.5	125.3	68.9	39.6	68.4	42.5
r	20	1	10	9.1	7.2	2.8	69.7	61.9	18.7	13.9	6.0	68.1	57.2	28.9	20.4	10.2	64.9	50.3	47.6	30.1	16.4	65.4	45.4
r	20	2	5	9.5	7.2	3.7	61.3	49.0	19.7	15.0	9.1	53.8	39.4	32.1	22.9	15.8	50.9	31.3	52.4	35.1	25.0	52.2	28.7
r	20	2	10	5.3	4.9	2.2	58.2	54.7	11.2	10.1	4.7	57.7	53.0	17.0	15.5	7.5	55.9	51.4	27.1	22.9	12.4	54.2	45.9
average				7.7	5.9	3.1	60.2	49.4	16.2	11.9	7.6	54.9	39.9	27.2	18.6	12.4	55.4	37.3	48.9	30.4	18.2	59.8	40.0
c	10	1	5	13.9	3.7	1.5	89.1	58.8	53.5	7.0	3.5	93.5	50.2	108.3	10.5	6.1	94.4	41.8	188.7	15.9	10.4	94.5	35.0
c	10	1	10	11.1	2.7	1.2	89.6	57.4	41.6	5.4	2.5	94.1	54.9	94.9	8.3	4.2	95.5	49.1	177.4	13.0	7.4	95.8	43.5
c	10	2	5	12.9	4.0	1.9	84.9	51.9	47.1	8.0	4.7	90.1	41.9	100.1	12.5	8.2	91.9	34.9	182.6	20.7	13.3	92.7	35.6
c	10	2	10	6.0	4.2	2.1	64.7	48.6	14.7	8.8	5.4	63.2	38.2	28.1	14.0	9.5	66.1	32.3	64.7	23.6	15.2	76.5	35.7
c	20	1	5	35.7	7.8	3.2	91.1	59.4	122.5	15.2	7.6	93.8	49.7	226.4	22.8	13.1	94.2	42.5	375.8	34.0	21.5	94.3	36.8
c	20	1	10	29.3	6.1	2.5	91.6	59.6	109.1	12.0	5.7	94.8	52.8	211.6	18.8	9.6	95.5	48.8	361.0	27.8	16.5	95.4	40.9
c	20	2	5	31.6	8.0	3.8	88.0	52.6	114.5	16.6	9.4	91.8	43.4	217.2	24.4	15.9	92.7	35.1	368.1	38.9	24.8	93.3	36.2
c	20	2	10	12.5	8.0	4.4	65.2	45.8	31.7	17.2	11.3	64.5	34.7	62.5	28.1	19.8	68.4	29.6	135.0	48.0	28.9	78.6	39.9
average				19.1	5.6	2.6	83.0	54.2	66.8	11.3	6.2	85.7	45.7	131.1	17.4	10.8	87.3	39.2	231.7	27.8	17.2	90.2	37.9

Table 5: Evaluation of the truncated brach-and-bound algorithm for problem classes with 200 customer orders and LKH heuristic

a	α	b	m	$k=3$					$k=5$					$k=7$					$k=10$				
				w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2	w_1	w_2	w_B	imp_1	imp_2
r	10	1	5	10.9	9.1	4.7	56.8	48.3	25.4	18.6	11.2	56.0	40.0	42.3	28.8	17.1	59.6	40.5	72.6	45.6	23.4	67.8	48.7
r	10	1	10	6.1	4.4	2.2	63.0	48.8	13.2	8.8	5.0	61.9	43.3	21.2	13.8	8.0	62.3	42.1	35.2	21.2	12.4	64.8	41.3
r	10	2	5	6.5	5.4	2.7	58.7	50.7	13.9	10.7	6.0	56.9	43.9	22.6	16.6	9.6	57.5	42.0	37.2	25.7	14.7	60.3	42.7
r	10	2	10	3.5	2.6	1.3	64.4	51.7	7.6	5.2	2.8	62.7	45.6	11.8	8.0	4.3	63.4	46.4	18.9	12.3	6.9	63.7	44.3
r	20	1	5	20.9	17.2	9.2	55.9	46.3	48.4	35.3	21.3	56.1	39.8	80.8	56.3	32.1	60.3	43.0	133.8	86.8	43.6	67.4	49.7
r	20	1	10	10.9	7.7	4.0	63.6	48.9	23.9	15.7	8.7	63.7	44.9	37.5	24.6	14.3	61.8	41.9	61.6	39.1	21.9	64.4	43.9
r	20	2	5	11.1	9.3	4.7	57.5	49.3	24.4	19.1	10.4	57.6	45.6	39.2	28.8	16.7	57.4	41.9	63.9	44.3	25.4	60.2	42.6
r	20	2	10	5.9	4.3	2.2	63.2	49.4	12.2	8.6	4.6	62.4	46.7	19.3	13.1	7.1	63.2	46.1	30.2	20.2	11.5	62.0	43.1
average				9.5	7.5	3.9	60.4	49.1	21.1	15.2	8.7	59.7	43.7	34.3	23.7	13.7	60.7	43.0	56.7	36.9	20.0	63.8	44.5
c	10	1	5	16.0	3.6	1.9	88.0	46.9	55.6	7.7	4.6	91.7	39.8	110.0	12.3	7.8	92.9	36.8	190.0	20.0	12.2	93.6	38.8
c	10	1	10	13.2	2.8	1.6	87.9	43.4	46.0	5.9	3.4	92.5	42.2	100.2	9.4	5.7	94.3	39.4	181.8	15.1	9.3	94.9	38.1
c	10	2	5	13.8	3.0	1.6	88.6	47.4	50.7	6.4	3.7	92.6	41.8	105.2	10.1	6.5	93.8	35.6	186.0	16.1	10.5	94.4	34.8
c	10	2	10	7.8	6.4	3.3	58.3	49.1	19.7	14.1	8.0	59.4	43.3	35.1	22.4	13.2	62.4	41.0	67.1	36.4	19.2	71.4	47.2
c	20	1	5	37.9	7.5	3.8	89.8	48.8	126.0	15.7	9.7	92.3	38.4	229.3	25.1	15.9	93.0	36.5	379.9	41.2	25.0	93.4	39.4
c	20	1	10	32.2	5.7	3.0	90.7	47.7	114.6	11.7	7.1	93.8	38.9	215.9	19.4	11.8	94.5	39.2	366.9	31.2	19.6	94.7	37.1
c	20	2	5	33.4	6.1	3.2	90.4	47.8	117.6	12.8	7.5	93.6	40.9	219.2	20.1	12.6	94.3	37.4	370.3	32.5	20.3	94.5	37.7
c	20	2	10	15.6	13.7	6.6	57.4	51.3	39.6	28.0	16.7	57.8	40.3	71.2	45.6	26.5	62.8	41.8	133.9	73.8	35.8	73.3	51.5
average				21.2	6.1	3.1	81.4	47.8	71.2	12.8	7.6	84.2	40.7	135.8	20.5	12.5	86.0	38.5	234.5	33.3	19.0	88.8	40.6

By application of the TBB algorithm, significant improvements regarding the total waiting time are achieved. In fact, the improvements range between 39.5% (S-shape, $N = 200$, $k = 5$, $a = r$, $\alpha = 10$, $b = 1$, $m = 5$) and 95.8% (S-shape, $N = 200$, $k = 10$, $a = c$, $\alpha = 10$, $b = 1$, $m = 10$). On average, the total waiting time can be reduced by 72.6%, which corresponds to a reduction of the processing time per customer order by 18 minutes. Thus, the results clearly demonstrate that using appropriate waiting instructions is pivotal for an efficient organization of the picking process.

In the following, the impact of the parameter settings on the amount of reduction obtained by application of the TBB algorithm is investigated. Regarding the number of customer orders and the pick-walk-time ratio, no effect on the amount of reduction can be identified. Concerning the routing algorithms, on average, the improvements obtained are also very similar (72.1% for the S-shape strategy and 73.1% for the LKH heuristic). However, the impact of the remaining parameters may be different depending on how the tours have been constructed.

Tours constructed by means of the S-shape strategy

Concerning the number of order pickers k, the amount of reduction rises with an increasing value of k if the S-shape strategy is used for the construction of the tours and if articles are assigned according to the class-based assignment policy. In fact, the average relative reduction of the total waiting time rises from 82.9% ($k = 3$) to 90.0% ($k = 10$). When assigning articles based on the random assignment policy, the largest improvements are observed in problem classes with 3 or 10 pickers.

Comparing the results for the two storage assignment policies, larger reductions are obtained in each problem class when articles are assigned following the class-based assignment policy. On average, the amount of reduction equals 57.9% and 86.4% for $a = r$ and $a = c$, respectively. If the class-based assignment policy is applied, the subaisles located near to the depot will be visited on almost every tour. Thus, the tours generated by means of the S-shape policy have a very similar structure, implying that order pickers traverse the subaisles in the same direction in most of the tours. In this case, the waiting times can be reduced significantly if other instructions are given than waiting at the entrance of the subaisle until the other picker has left this aisle. Furthermore, based on approach A_1, order pickers may often wait although no blocking situation occurs. If the random assignment policy is used, it is very likely that the sets of subaisles to be visited are significantly different for different tours, increasing the probability of order pickers traversing a subaisle in different directions at the same time. In this case, in both approaches, a picker has to wait until the other picker has left the subaisle. The only possibility of the TBB algorithm to improve the solution consists in the selection of the picker. Thus, the amount of reduction is smaller if the random storage assignment policy is used.

When considering the size of the warehouse for problem classes with $a = c$, the smallest improvements can be observed if the picking area consists of 2 blocks and 10 picking aisles. This can be explained by the fact that this is the only picking area where not all articles from the classes A and B have been assigned to subaisles of the first block. Thus, it is likely that the second block has to be visited, which results to more diverse tours (see also Subsection 4.3). With the same line of argumentation as for the impact of the storage assignment policy on the amount of reduction, smaller amounts of improvement can be justified in this case.

Tours constructed by means of the LKH heuristic

If the tours have been constructed by application of the LKH heuristic, the amount of reduction increases with an increasing number of order pickers for problem classes with $a = c$. This coincides with the observations related to problem classes where the S-shape strategy has been used. The relative reduction of the total waiting time amounts to 81.5%, 83.9%, 85.8% and 88.7% if 3, 5, 7 and 10 pickers are available, respectively. For $a = r$, the relative reduction obtained is similar for problem classes with 3, 5 and 7 pickers, while the largest reductions can be observed in classes with 10 pickers.

Regarding the storage assignment policy, larger relative reductions can be identified for the class-based assignment policy (85.0%) than for the random assignment strategy (61.3%). The only problem classes where the amount of reduction is larger for $a = r$ are characterized by a picking area containing 2 blocks and 10 picking aisles. At the same time, these problem classes represent the classes with the smallest relative reductions obtained if the class-based assignment procedure has been applied. This result matches with the corresponding observation for classes with $a = c$ and tours being constructed by means of the S-shape strategy.

6.2.2 Improvements by selecting the picker who has to wait

The results presented above show that the waiting instructions have a large impact on the total waiting time. In approach A_2 and in the TBB algorithm, identical waiting instructions are given if the same order picker has to wait. The difference between these approaches can only be found in the selection of the picker to whom waiting instructions are given for a certain blocking situation. While the picker who left the depot at a later point in time will always wait according to A_2, the selection of the picker is dependent on the effect of the decision on the current and future blocking situations in the TBB algorithm.

The amount of reduction ranges from 12.4% (S-shape, $N = 200$, $k = 5$, $a = r$, $\alpha = 10$, $b = 1$, $m = 5$) to 61.9% (S-shape, $N = 200$, $k = 3$, $a = r$, $\alpha = 20$, $b = 1$, $m = 10$). As expected, the average relative reduction is smaller than the improvements achieved regarding A_1. However, compared to approach A_2, the TBB algorithm can reduce the total waiting time by 43.2% on average, which shows that the selection of the picker who has to wait also represents an important decision, having a significant impact on the resulting total waiting time.

The reductions obtained for varying numbers of customer orders, pick-walk-time ratios and routing algorithms are of similar magnitude, respectively. Regarding the number of order pickers k, it can be observed that an increasing number of pickers results in smaller relative reductions. While relative reductions of 50.0% are achieved for $k = 3$, the total waiting time can be decreased by 41.1% for $k = 10$. This can be explained by the fact that the total waiting time significantly rises with an increasing number of order pickers. With respect to the absolute reduction of the total waiting time, average improvements of 10.3 hours are obtained for problem classes with 10 pickers, where the waiting time is reduced by 2.4 hours only if 3 pickers are available. The impact of the assignment policy and the size of the picking area is dependent on the underlying routing algorithm. Whereas larger relative reductions can be observed for the class-based assignment policy if tours are constructed by means of the S-shape strategy (42.0% for $a = r$ and 44.1% for $a = c$), the opposite holds for problem classes where the LKH heuristic has been applied (44.9% for $a = r$ and 41.7% for $a = c$). Concerning the size of the warehouse, it can be seen that the number of blocks and the number of picking aisles do not affect the amount of reduction obtained if problem classes based on the LKH heuristic are considered. For classes in which the S-shape strategy has been applied, the average relative reduction drops with a decreasing number of picking aisles if articles have been assigned according to the random assignment policy. Otherwise, the least reductions are obtained in case of a picking area including 2 blocks and 10 picking aisles.

As can be seen from the results of the numerical experiments, both the selection of the order picker who has to wait and the waiting instructions actually given to the respective pickers have a strong impact on the total waiting time. By carefully dealing with both types of decisions, the TBB algorithm manages to reduce the total waiting time significantly, which has a positive effect on the processing times of the orders. In the following subsection, the TBB algorithm is evaluated with respect to the computing time required in order to investigate whether this approach is suitable for dealing with large-sized instances arising in practical applications.

6.3 Computing times

The computing times required by approaches A_1 and A_2 are below one second and, therefore, they are negligible. Regarding the TBB algorithm, for each problem class, the average computing time (in seconds) is depicted in Table 6. As can be seen from the table, the average computing time is below one minute for each problem class. Thus, it can be concluded that the TBB algorithm is suitable for dealing with very large instances as well. More precisely, the average computing time required for solving an instance by means of the TBB algorithm ranges from 0.1 seconds (several classes with S-shape, $N = 100$ and $k = 3$) to 56.9 seconds (S-shape, $N = 200$, $k = 10$, $a = c$, $\alpha = 20$, $b = 2$, $m = 10$).

Table 6: Computing times [seconds] required by the truncated branch-and-bound algorithm

N	a	α	b	m	$k=3$		$k=5$		$k=7$		$k=10$	
					S-shape	LKH	S-shape	LKH	S-shape	LKH	S-shape	LKH
100	r	10	1	5	0.1	0.2	0.2	0.4	0.5	1.0	2.0	3.7
100	r	10	1	10	0.3	0.3	0.5	0.6	0.9	1.0	1.5	2.2
100	r	10	2	5	0.2	0.3	0.4	0.6	0.7	1.3	1.4	2.7
100	r	10	2	10	0.4	0.5	0.9	0.9	1.7	1.5	3.0	2.6
100	r	20	1	5	0.1	0.2	0.2	0.5	0.6	1.3	2.7	6.2
100	r	20	1	10	0.2	0.3	0.6	0.7	1.0	1.2	2.3	2.4
100	r	20	2	5	0.2	0.3	0.5	0.8	0.8	1.4	1.6	3.4
100	r	20	2	10	0.5	0.5	1.1	1.0	1.9	1.7	3.7	3.2
100	c	10	1	5	0.1	0.1	0.2	0.3	0.3	0.5	0.6	1.1
100	c	10	1	10	0.1	0.2	0.3	0.4	0.4	0.7	0.8	1.2
100	c	10	2	5	0.2	0.2	0.4	0.4	0.8	0.8	2.1	1.4
100	c	10	2	10	0.5	0.6	1.0	1.3	1.8	2.3	5.4	5.8
100	c	20	1	5	0.1	0.1	0.2	0.3	0.4	0.6	0.8	1.6
100	c	20	1	10	0.2	0.2	0.3	0.5	0.7	0.9	1.2	1.5
100	c	20	2	5	0.3	0.2	0.5	0.6	1.0	0.9	2.7	2.0
100	c	20	2	10	0.5	0.7	1.1	1.6	2.5	3.3	8.3	10.1
200	r	10	1	5	0.5	0.8	0.9	2.4	2.2	5.5	8.4	22.7
200	r	10	1	10	0.9	1.3	2.1	2.7	3.7	4.7	6.7	9.2
200	r	10	2	5	1.0	1.4	2.1	3.3	3.4	6.4	6.1	12.9
200	r	10	2	10	1.8	2.2	4.5	4.2	7.5	6.9	14.5	12.4
200	r	20	1	5	0.5	0.9	1.1	2.6	2.5	7.1	10.9	40.2
200	r	20	1	10	1.1	1.3	2.7	3.1	4.8	5.8	9.7	11.2
200	r	20	2	5	1.1	1.5	2.6	3.7	4.2	7.0	8.2	16.8
200	r	20	2	10	2.0	2.3	5.0	4.8	8.6	7.8	17.8	14.4
200	c	10	1	5	0.4	0.5	0.8	1.3	1.3	2.4	2.4	6.1
200	c	10	1	10	0.7	1.0	1.4	2.0	2.2	3.0	3.7	5.9
200	c	10	2	5	1.1	1.0	2.4	2.3	4.4	3.7	9.9	7.2
200	c	10	2	10	2.0	2.7	4.6	5.8	8.6	11.2	26.3	29.1
200	c	20	1	5	0.4	0.6	1.1	1.5	1.7	2.9	3.4	9.7
200	c	20	1	10	0.9	1.1	1.9	2.4	3.1	3.9	5.9	7.7
200	c	20	2	5	1.2	1.2	2.8	2.8	5.4	4.7	15.1	9.3
200	c	20	2	10	2.4	3.2	5.7	7.1	12.6	16.3	45.9	56.9

The number of nodes included in the tree has a major impact on the computing time. Since nodes are generated after a blocking situation has been identified, it can be expected that the largest computing times are required for solving instances from problem classes where order pickers tend to block each other quite often. In fact, computing times rise with an increasing number of order pickers and an increasing pick-walk-time ratio. Furthermore, larger computing times can be observed when tours have been constructed according to the LKH heuristic. This can also be explained by the number of blocking situations arising as, on average, 214 blocking situations are considered for the LKH heuristic while only 188 blocking situations occur for the S-shape strategy. Therefore, if the S-shape strategy has been used for the generation of the tours, the average number of nodes created in the TBB algorithm is lower (1260 nodes for the S-shape strategy compared to 1792 nodes for the LKH heuristic), leading to smaller computing times. The number of customer orders represents another parameter that has an impact on the number of blocking situations. The more customer orders are to be processed, the more tours are to be performed, resulting in a larger number of blocking situations in total. Thus, it is not surprising that computing times increase (from 1.2 seconds to 6.1 seconds) if 200 instead of 100 orders are considered.

A significant part of the computing time is also spent on the identification of blocking situations. Whether a blocking situation arises, is checked each time before a picker enters a subaisle. The more subaisles are to be entered in a tour, the more checks have to be performed. Thus, the identification of blocking situations is more time-consuming in case of large picking areas including many subaisles, leading to slightly higher computing times for increasing numbers of blocks and picking aisles.

7 Conclusion and outlook

In this paper, we dealt with the problem of guiding order pickers through a picking area including narrow subaisles. In narrow subaisles, order pickers can neither pass nor overtake each other. Thus, an order picker may have to wait until another picker has completed the operations in a certain subaisle. Although it is known that, in particular when many order pickers are employed, the arising waiting times have a significant negative impact on the efficiency of the picking process, waiting times are rarely taken into account when guiding order pickers.

In the first part of the paper, by means of numerical experiments, settings are identified where the proportion of the total waiting time as part of the total processing time is quite large and situations are pointed out where waiting times can be neglected. For the determination of the total waiting time, two different approaches are designed in which the decisions regarding the pickers who have to wait are made based on suggestions from the literature. The results of the experiments show that the consideration of waiting times is inevitable for an efficient organization of the picking process, as the proportion of the total waiting time amounts up to 62%, i.e. almost two-thirds of the total processing time is spent on waiting. In order to reduce waiting times, in the second part of the paper, a truncated branch-and-bound algorithm is proposed where blocking situations are identified chronologically and nodes are generated according to decisions regarding the selection of the picker who has to wait in the respective blocking situation. By means of numerical experiments, it is demonstrated that this algorithm leads to excellent results within very short computing times. It is pointed out that waiting times can be decreased by up to 96% if more sophisticated waiting instructions are used instead of instructing the pickers to wait at the entrance of the subaisle until no other picker is in this subaisle. Furthermore, it is shown that reductions of up to 62% can be obtained by simply putting more emphasis on the selection of the picker who has to wait in a certain situation.

It has to be noted that all considerations in this paper are based on the assumption that the travel velocity of all pickers is constant and both the travel velocity and the pick times are deterministic. This is a standard assumption in the literature. However, this assumption is very critical as it is hardly met in practice. First, human operators do not travel with a constant velocity. They have to accelerate after having performed the operations at a location and they decelerate before stopping at a pick location or when switching between picking aisles. Moreover, the travel velocity may differ regarding the travel directions (e.g. the velocity may be lower when an order picker returns to a cross aisle by backing). Second, the travel velocity is not only varying but also stochastic in practice. This also holds for the pick time because a human operator does not need exactly the same amount of time each time he/she performs a certain operation. Thus, the integration of varying or even stochastic travel velocities and pick times represents a very important area of future research.

Further research could also concentrate on the extension of the waiting instructions. In the truncated branch-and-bound approach, pickers either wait or they perform their operations as planned. For the reduction of the waiting times, it could also be advantageous that order pickers deviate from their paths. For example, if a subaisle is traversed without retrieving an item, another subaisle could be chosen (Chen et al., 2016). Moreover, the tours could be modified completely, i.e. the sequence according to which the items are to be picked could be changed. In both scenarios, the minimization of the total waiting time would not represent a valid objective and the total processing time should be used for the evaluation of solutions instead.

The consideration of the assignment of customer orders to the pickers and the sequencing according to which customer orders are to be processed represents another promising topic for future research. These decisions provide much more flexibility, which can be expected to prevent many blocking situations from arising. Another interesting aspect can be found in the integration of the Order Batching Problem, i.e. customer orders can be grouped into batches and then processed on a single tour. It can be expected that the batching of customer orders leads to an increase of the proportion of the total waiting time (Gue et al., 2006; Hong et al., 2010). However, it will significantly reduce the total processing time.

References

Bozer, Y. A. & Kile, J. W. (2008): Order batching in walk-and-pick order picking systems. *International Journal of Production Research* 46, 1887-1909.

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, 77-85.

Chen, F.; Wang, H.; Xie, Y. & Qi, C. (2016): An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse. *Journal of Intelligent Manufacturing* 27, 389-408.

de Koster, R.; Le-Duc, T. & Roodbergen, K. J. (2007): Design and control of warehouse order picking: A literature review. *Science Direct* 182, 481-501.

Gue, K. R.; Meller, R. D. & Skufca, J. D. (2006): The effects of pick density on order picking areas with narrow aisles. *IIE Transactions* 38, 859-868.

Hall, R. W. (1993): Distance approximations for routing manual pickers in a warehouse. *IIE Transactions* 25, 76-87.

Helsgaun, K. (2000): An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research* 126, 106-130.

Henn, S.; Koch, S.; Dörner, K. F.; Strauss, C. & Wäscher, G. (2010): Metaheuristics for the order batching problem in manual order picking systems. *BuR - Business Research* 3, 82-105.

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

Henn, S. (2015): Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. *Flexible Services and Manufacturing* 27, 86-114.

Hirschberg, C. (2015): Variantenvielfalt oder was Automobile und Gurken gemeinsam haben. URL: http://ipl-mag.de/ipl-magazin-rubriken/scm-daten/390-variantenvielfalt

Ho, Y.-C. & Chien, S.-P. (2006): A comparison of two zone-visitation sequencing strategies in a distribution centre. *Computers & Industrial Engineering* 50, 426-439.

Hong, S.; Johnson, A. L. & Peters, B. A. (2010): Analysis of picker blocking in narrow-aisle batch picking. *Progress in Material Handling Research: Proceedings of 2010 International Material Handling Research Colloquium*, Ellis, K.P.; Gue, K.; de Koster, R.; Meller, R.; Montreuil, B.& Ogle, M. (Eds.). Charlotte: The Material Handling Institute.

Jarvis, J. M. & McDowell, E. D. (1991): Optimal product layout in an order picking warehouse. *IIE Transactions* 23, 93102.

Le-Duc, T. & de Koster, R. (2007): Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operational Research* 176, 374-388.

Pan, J. C.-H. & Shih, P.-H. (2008): Evaluation of the throughput of a multiple-picker order picking system with congestion consideration. *Computers & Industrial Engineering* 55, 379-389.

Pan, J. C.-H. & Wu, M.-H. (2012): Throughput analysis for order picking system with multiple pickers and aisle congestion considerations. *Computers & Operations Research* 39, 1661-1672.

Parikh, P. J. & Meller, R. D. (2009): Estimating picker blocking in wide-aisle order picking systems. *IIE Transactions* 41, 232-246.

Parikh, P. J. & Meller, R. D. (2010): A note on worker blocking in narrow-aisle order picking systems when pick time is non-deterministic. *IIE Transactions* 42, 392-404.

Petersen, C. G. & Schmenner, R. W. (1999): An evaluation of routing and volume-based storage policies in an order picking operation. *Decision Science* 30, 481-501.

Rakrouki, M. A.; Ladhari, T. & Tkindt, V. (2012): Coupling genetic local search and recovering beamsearch algorithms for minimizing the total completion time in the single machine scheduling problem subject to release dates. *Computers & Operations Research* 39, 1257-1264.

Ratliff, H. D. & Rosenthal, A. R. (1983): Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research* 31, 507-521.

Roodbergen, K. J. (2001): Layout and routing methods for warehouses. Trial: Rotterdam.

Roodbergen, K. J. & de Koster, R. (2001): Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research* 133, 32-43.

Skufca, J. D. (2005): k workers in a circular warehouse: A random walk on a circle, without passing. *Society for Industrial and Applied Mathematics* 47, 301-314.

Statista (2016): Jährliche Anzahl der Bestellungen bei Zalando in den Jahren 2011 bis 2015. URL: https://de.statista.com/statistik/daten/studie/325862/umfrage/anzahl-aller-bestellungen-bei-zalando-pro-jahr/

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, 755-763.

Tompkins, J. A.; White, J. A.; Bozer, Y. A. & Tanchoco, J. M. A. (2010): Facilities planning. 4th edition, John Wiley & Sons, New Jersey.

Van Nieuwenhuyse, I. & de Koster, R (2009): Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics* 121, 654-664.

Yu, M.; de Koster, R. (2009): The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research* 198, 480-490.

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274