Transportation Research Part E 102 (2017) 38–59

Contents lists available at ScienceDirect

Transportation Research Part E

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

Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses

Arjan S. Dijkstra *, Kees Jan Roodbergen

University of Groningen, Dept Operations, PO Box 800, 9700 AV Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 27 October 2016
Received in revised form 28 February 2017
Accepted 4 April 2017

Keywords:
Warehouse management
Storage assignment
Dynamic programming

ABSTRACT

Order picking is one of the most time-critical processes in warehouses. We focus on the combined effects of routing methods and storage location assignment on process performance. We present exact formulas for the average route length under any storage location assignment for four common routing methods. Properties of optimal solutions are derived that strongly reduce the solution space. Furthermore, we provide a dynamic programming approach that determines storage location assignments, using the route length formulas and optimality properties. Experiments underline the importance of the introduced procedures by revealing storage assignment patterns that have not been described in literature before.

1. Introduction

Warehouses offer a variety of benefits to the company employing them, among which accomplishing least total cost logistics with a desired level of customer service and supporting the just-in-time programs of suppliers and customers (De Koster et al., 2007). However, operating a warehouse can be quite costly as typically a great deal of manual labor is involved. One of the most laborious and costly activities in a warehouse is order picking; the process of retrieving a set of items from storage locations in response to customer orders. The cost of this process can be as much as 55% of the total operating costs of a warehouse (Tompkins et al., 2003). Several operational decisions strongly influence the performance of the order-picking process.

Before customer orders can be retrieved, items must first have been stored in the available locations. The problem of choosing appropriate storage locations for the items is called the storage location assignment problem (Hausman et al., 1976; Gu et al., 2007). This problem shows similarities with the Assignment Problem, a fundamental problem from the field of Operations Research. Both problems require the matching of objects from two mutually exclusive sets, in our case a matching of items to locations. However, the two problems differ strongly in their objective functions. The objective function of the standard Assignment Problem has a simple structure, which makes the problem solvable in polynomial time. The storage location assignment problem, on the other hand, has the objective to minimize the average route length traveled by the workers (order pickers) while retrieving items from locations in the warehouse. This results in a complex function that depends on the layout of the warehouse, the routing method employed, the demand frequencies of all items, and the item-to-location assignment itself.

* Corresponding author.
E-mail addresses: a.s.dijkstra@rug.nl (A.S. Dijkstra), k.j.roodbergen@rug.nl (K.J. Roodbergen).

http://dx.doi.org/10.1016/j.tre.2017.04.003
1366-5545/

The routing method prescribes how an order picker should navigate the warehouse to collect the items in an order. A good routing method yields short route lengths, which decreases the time to pick an order. The physical structure of the warehouse limits the movements of the order picker, who has to navigate between the racks that store the items. Essentially, the routing problem in warehouses classifies as a special case of the Steiner Traveling Salesman Problem (see De Koster et al., 2007), that in some layouts is solvable to optimality in polynomial time (Ratliff and Rosenthal, 1983). However, as De Koster et al. (2007) note, "in practice, the problem of routing order pickers in a warehouse is mainly solved by using heuristics". For this purpose, a number of heuristic routing methods are available from literature. These heuristics can be of a constructive (greedy) nature, employing properties of the layout to find solutions (Hall, 1993) or contain a procedure to search the solution space (Theys et al., 2010).

In this paper, we develop methods for determining storage location assignments in warehouses where order pickers span multiple aisles in each route, and where each order may contain any number of items to be picked (*multi-aisle multi-item picking*). We solve a number of multi-aisle multi-item storage location assignment problems and can guarantee optimality for the first time. As a core component for achieving this, we present formulas for four routing methods that give the value for expected route length in multi-aisle multi-item picking warehouses. In contrast to existing research work, these formulas give the exact expected route length, not an approximation, and they hold for any storage location assignment. Furthermore, structural properties of optimal solutions are derived, which aid in narrowing down the solution space. Due to the low computational requirements for our formulas, in combination with the structural properties, a number of instances can be solved to proven optimality by means of complete enumeration. Using the formulas at its core, we present a dynamic programming approach for determining storage location assignments. The dynamic program provides optimal solutions for one routing method, and near-optimal solutions for another routing method. For the remaining two routing methods, we have no optimal benchmark, but the dynamic program is shown to consistently outperform common storage location assignment rules from literature. Results from our experiments demonstrate patterns for storage location assignment that have not been described before in literature, and that challenge current assumptions about predetermined storage location assignment patterns. A detailed comparison of our work to existing literature is given in Sections 3 and 4.

The remainder of this paper is organized as follows. First, in Section 2 the warehouse layout and the routing methods used are explained. Thereafter, the backgrounds on storage location assignment and on route length estimation are given in Sections 3 and 4, respectively. In Section 5 the problem is defined mathematically. In Section 6 we derive formulas for the exact expected route lengths for four routing methods. These formulas are used to determine optimality conditions in Section 7 and serve as inspiration for the Dynamic Program presented in Section 8. Section 9 provides computational results. Finally, concluding remarks are given in Section 10.

2. Warehouse layout and routing methods

We consider a warehouse with two cross aisles as depicted in Fig. 1. The warehouse has a number of parallel aisles where items are stored in locations. A front cross aisle and a back cross aisle provide access to the aisles, but do not contain storage locations themselves. All routes start and end at the depot (marked "D" in Fig. 1, where the order picker takes an empty

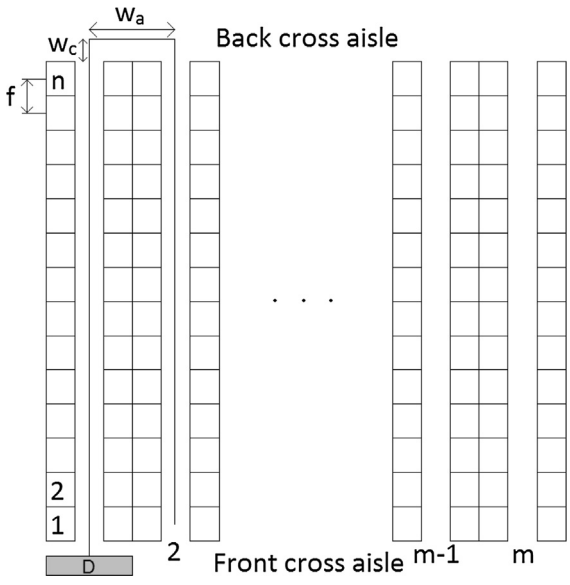


Fig. 1. Schematic overview of the warehouse.

product carrier at the start of the route, and deposits picked items at the end of the route. The depot is located at the left of the front cross aisle. The aisles are sufficiently narrow for order pickers to be able to retrieve items from both sides of an aisle without moving sideways, implying that for the routing we can assume order pickers to travel in the exact middle of the aisles. Each item uses identical storage space. This is a common configuration that is studied widely in literature (e.g., Chen et al., 2010; Chew and Tang, 1999; Le-Duc and De Koster, 2005; Petersen and Schmenner, 1999; Rao and Adil, 2013b). In this paper, we consider the configuration to be fixed. Examples of papers studying warehouse design are Heragu et al. (2005), Hsieh and Tsai (2005), Hwang and Cho (2006), Parikh and Meller (2010), Thomas and Meller (2014).

Even though our primary goal is to optimize the storage location assignment, a significant part of the paper deals with routing methods. This is for two reasons. First, the storage location assignment is strongly dependent on the routing method employed (Petersen and Schmenner, 1999). Therefore, we aim for optimizing the storage location assignment for each of four constructive routing methods. Second, the quality of the mathematical description of the routing method's expected route length influences the quality of the storage location assignment optimization process. After all, if we cannot precisely determine the expected route length of any given configuration, then comparisons between configurations cannot be reliably made by any optimization procedure. We focus on four constructive routing methods that appear regularly in literature and practice. Examples of routes for each of these methods are depicted in Fig. 2.

For the *Return routing method*, the order picker enters each aisle from the front, if at least one item has to be picked in that aisle. The aisle is traversed up to the pick item farthest from the front cross aisle, after which the order picker returns to the front cross aisle. Connections between aisles are exclusively made via the front cross aisle. This is the only logical routing method for warehouses that have a single cross aisle. It is used in many articles, including Caron et al. (1998), Le-Duc and De Koster (2005), and Chen et al. (2010). A method that does use both cross aisles, is the S-shape (or transversal) method. For the *S-shape routing method*, starting from the depot, the order picker visits aisles from left to right. Each aisle in which a pick item is located, is traversed completely, implying that an aisle entered from the front cross aisle is exited via the back cross aisle, and vice versa. In the case that the number of aisles with pick items is odd, the last aisle does not have to be traversed completely; instead, the picker can return to the front cross aisle after picking the last item. Finally, the order picker returns to the depot. Articles that consider S-shape routing include Chen et al. (2010), Pan et al. (2012), and Rao and Adil (2013a).

The third and fourth method we consider are the *largest-gap* and *midpoint routing methods*. Both methods follow the perimeter of the warehouse. The first aisle with pick items is entered and traversed completely. From the back cross aisle, each subsequent aisle is entered and left from the back. The last aisle with pick items is traversed completely to return to the front cross aisle. Finally, the order picker travels through the front cross aisle towards the depot while entering and leaving each aisle from the front. This implies that, except for the first and last aisles with pick items, all aisles may be visited twice. The distinction between the midpoint and the largest-gap routing method is in the selection of items to pick when entering the aisle from respectively the front or back cross aisle. For the midpoint routing method, items that are in the back half of the aisle are picked via the back cross aisle, and the items in the front half of the aisle via the front cross aisle. The middle of the warehouse is therefore never crossed by the order picker, except in the first and last aisle with pick items. For the largest-gap routing method, the division of pick items is optimized, i.e., the largest stretch of the aisle that does not contain pick items, is not visited. Midpoint and largest-gap routing methods are used in, e.g., Hall (1993), Petersen and Schmenner (1999), and Dekker et al. (2004).

Note that for all four routing methods, aisles that contain no pick items are not visited. Furthermore, in case an order consists of only items in a single aisle, then all routing methods give the same route as the Return routing method.

3. Background on storage location assignment

Storage assignment problems can be subdivided into the problem of assigning items to storage departments, assigning items to zones within these departments, and assigning items to locations within these zones (Gu et al., 2007).

The assignment of items to departments, such as reserve storage and forward picking areas, can be guided by external factors, such as departments dedicated to a single customer. However, items need not be assigned to a single department. Frequently, items are assigned to both a reserve area, as well as a forward picking area. The reserve area is designed to hold large quantities of products, whereas the forward picking area is designed for fast retrieval of products in response to customer orders. The forward-reserve problem then is how inventory of an item should be divided between these departments (e.g., Frazelle et al., 1994). Extensions of this problem study, for example, how cross-docking can be incorporated (Heragu et al., 2005).

Physical attributes of items are usually taken into account when assigning items to zones within departments. Storage technology and physical arrangement are usually decided on this level (Gu et al., 2007). Additionally, zones can be employed to organize order-picking activities more efficiently (see, e.g. Jewkes et al., 2004).

Finally, the assignment of items to storage locations within zones aims to minimize the expected time of retrieval of these items. This is what we refer to as the *storage location assignment problem* in this paper. Often papers on storage location assignment restrict the solution space to class-based storage. Under *class-based storage*, all items are divided into a number of (often two or three) classes based on demand frequency of the items. The fastest moving items are usually called A-items. The next fastest moving items are called B-items, and so on. Each class needs to be assigned to a part of the warehouse, but

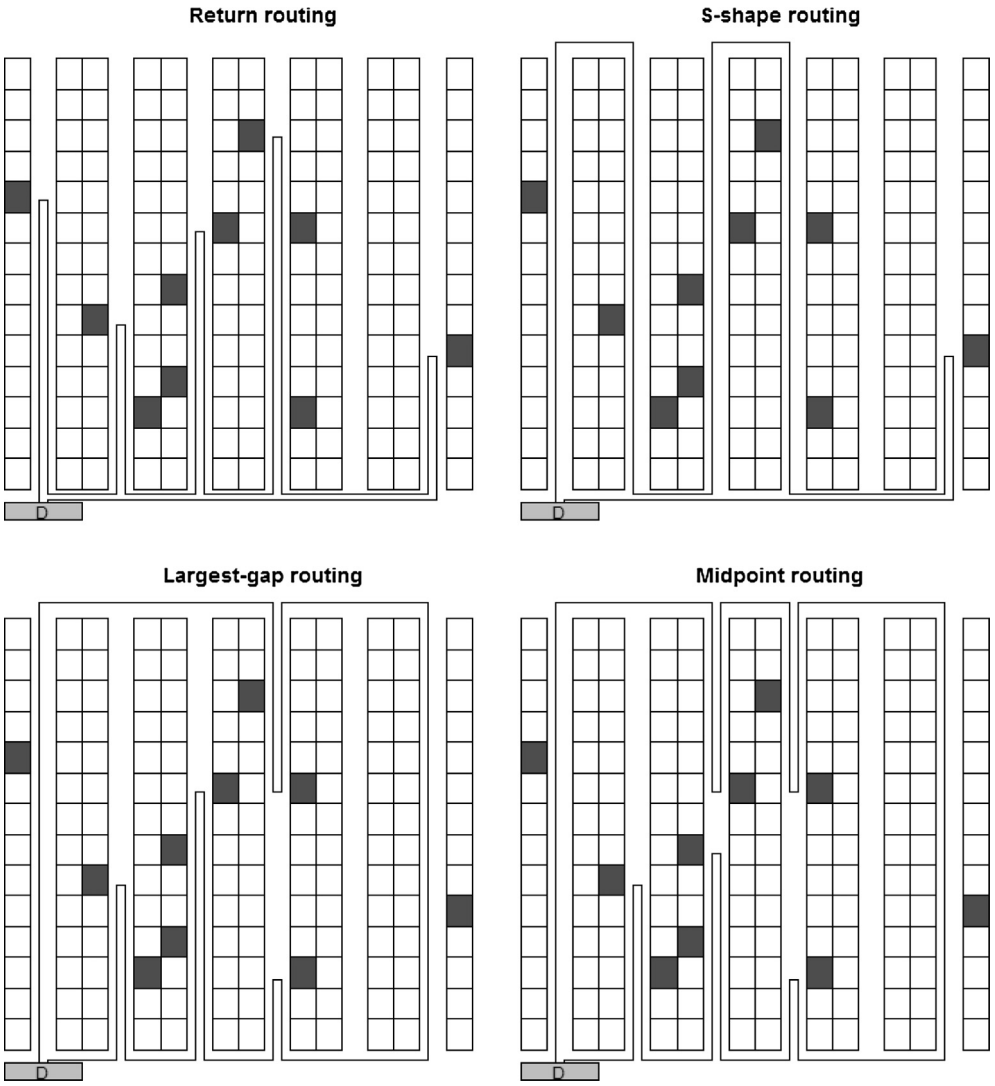


Fig. 2. Example routes for the four different routing methods. A solid square indicates a pick location.

assignment within the class is assumed random with equal probability. Demand frequencies for all items within a class are subsequently assumed equal to the average of the class to mimic long-term behavior. As a consequence, for a warehouse with m aisles each with n locations, instead of $(mn)!$ possible storage configurations, only $(mn)!/\prod_{i=1}^{k}(t_i!)$ unique configurations remain, where k denotes the number of classes and t_i is the number of locations assigned to class i. It must be noted that the remaining number configurations is usually still a very large number for most practical instances, causing complete enumeration to be intractable. To counter this problem a common approach for studies on storage location assignment for multi-aisle multi-item picking is to introduce pre-determined storage patterns, and subsequently to compare performance of these patterns, often in combination with other operating policies. Commonly used patterns include within-aisle storage and across-aisle storage (Petersen and Schmenner, 1999). These studies provide valuable insights into the issues of storage location assignment, however, it typically gives no information on the absolute quality of the solutions. Research includes Caron et al. (1998), Petersen and Schmenner (1999), Hwang et al. (2004) and Chen et al. (2010).

Research on storage location assignment that aims at developing procedures that can actively search for a good storage location assignment is very limited for multi-aisle multi-item picking. Dekker et al. (2004) use a 2-opt exchange procedure based on Lin and Kernighan (1973) to improve on a number of predetermined class-based storage configurations. For a different layout, Le-Duc and De Koster (2005) also apply 2-opt to improve the storage location assignment. Related work of Pan et al. (2015) presents a genetic algorithm for storage location assignment in a pick-and-pass system (single-aisle multi-item picking). Muppani and Adil (2008) present a branch-and-bound method for finding a storage location assignment for unit load warehouses (multi-aisle single-item picking). Ene and Öztürk (2011) use an ILP to determine the storage locations in

a warehouse from the automotive industry with 7 aisles and a single cross aisle, where travel time is approximated by a weighting factor for different storage classes.

To our knowledge no optimal algorithm has been presented in literature for the problem of determining storage location assignments in multi-aisle multi-item picking, nor have any instances been solved to proven optimality for any routing method. We present in this paper a constructive approach based on dynamic programming, which yields proven optimal solutions in polynomial time for the return routing method. Furthermore, heuristic procedures for constructing solutions for multi-aisle multi-item picking are scarce in literature, and their performance is not exactly known, since solutions have not been benchmarked against optimal solutions. We use our dynamic programming approach for the S-shape routing method to obtain near-optimal solutions, as is proven by a comparison to benchmark instances that we could solve to optimality due to derived properties. Finally, application of our method with two other routing methods reveals new patterns for storage location assignment that have not been reported before in literature, and may provide new cues for further research towards optimality.

4. Background on route length estimation

To assess the efficiency of any storage location assignment, we need to be able to evaluate the resulting average route length. The two main methods to achieve this are the creation of a simulation model and developing formulas based on statistical properties of the routing method. We focus on the latter, since we aim towards optimality.

To the best of our knowledge, no formula exists that gives the exact average route length for any routing method in a multi-aisle multi-item picking warehouse. Formulas either contain statistical approximations, are based on a simplified version of the actual routing method, or both. For return routing, Le-Duc and De Koster (2005) assume the storage locations in an aisle are continuous to obtain an approximation of expected route length under class-based storage. Another approximation for return routing is proposed by Caron et al. (1998), who assume the number of pick items per aisle to be equal to the total number of pick items divided by the number of aisles and use order statistics to determine the expected travel distance in an aisle. This method only allows for identical storage in all aisles. The methods both yield maximum differences with simulation of about 4%. Rao and Adil (2013b) assume all pick items are distributed over the classes per fraction of their total turnover, which yields an approximation for the travel in a single aisle with flexible class boundaries.

For S-shape routing, approximations for average route length mostly differ in the way they handle the return in the last aisle when the number of aisles is odd. Hall (1993) adds the length of one aisle with a probability of 0.5 regardless of the actual probability and travel distance in the last aisle. Jarvis and Mcdowell (1991) assume the picker continues to the back of the warehouse and do not account for the return to the front cross aisle, which essentially provides the same distance as in Caron et al. (1998) where it as assumed that the return is always from the middle of the last aisle. An exact expression for average route length under class-based storage is given by Chew and Tang (1999), but this allows only one storage class to be stored in a single aisle. They derive an approximation based on their exact formula, which has been improved by Roodbergen and Vis (2006) for random storage and by Rao and Adil (2013a) for class-based storage.

For midpoint and largest-gap routing, the approximation of average route length has received less attention in literature. Two different approximations for midpoint routing under random storage are proposed by Hall (1993). Additionally, Hall (1993) derived an approximation for largest-gap routing with a component in the formula that contains results from a simulation of the largest gap in an aisle for a given number of pick items. Roodbergen and Vis (2006) improved this approximation. Hwang et al. (2004) propose a travel model for midpoint routing for a specific storage strategy, based on approximate probabilities to enter an aisle as proposed by Kunder and Gudehus (1975). In this paper, we present exact formulas for the expected route length for all previously mentioned routing heuristics, i.e., return, S-Shape, largest-gap, and midpoint routing.

5. Problem formulation

We consider an order picking area as described in Section 2 with m aisles each with n locations. In line with literature on storage assignment and warehouse configuration, we assume both sides of the aisle are merged into a single column for simplicity of exposition (e.g., Le-Duc and De Koster, 2005; Parikh and Meller, 2010). This assumption is not limiting for our analysis, which will be addressed in the relevant sections. Define K to be the set of items that need to be assigned to storage locations. Define for each $k \in K$ a random variable $X_k \in \{0,1\}$ with $X_k = 1$ corresponding to the event that item k is demanded on a given order, and $X_k = 0$ otherwise. Since we assume demand for items to be independent, X_k are independent Poisson trials with probability $p_k \equiv P(X_k = 1),\ \forall k \in K$. The demand distribution is completely described by the set $\{p_k, k \in K\}$. In the context of storage location assignment this demand model is also used in, for example, Jarvis and Mcdowell (1991) and Eisenstein (2008). Given a realization of the X_k's, we define an order as $O \equiv \{k | k \in K, X_k = 1\}$. We assume all items can be stored in any storage location, as is common in literature (e.g., Le-Duc and De Koster, 2005; Eisenstein, 2008; Pan et al., 2012). Let A be a $m \times n$ matrix denoting a storage location assignment, where a_{ij} gives the index of the item assigned to location j in aisle i, and $a_i = (a_{i1}, \ldots, a_{in})$ denotes all assignments within an aisle i. For ease of notation we define $p_{ij} \equiv P(X_{a_{ij}} = 1)$, that is, if item k is assigned to location j in aisle i then $p_{ij} = p_k$. Note that we assume K has size $m \times n$. Often the set of items to be stored is less than $m \times n$. In this case, dummy items with picking probability 0 can be added to meet the assumption.

The way the model is defined now does not exclude the possibility that an order contains no pick items (an *empty order*). Therefore, we condition on the event that the order has at least one pick item. Similar to Eisenstein (2008), define $\mathscr{P}$ to be the probability that an order is non-empty:

$$\mathscr{P} \equiv P(|O| > 0) = 1 - \prod_{k \in K}(1 - p_k),$$

where $|O|$ denotes the number of elements in O. We assume the capacity of the order picker is sufficient to fulfill any order, as is common in literature (see, e.g., Jarvis and Mcdowell, 1991; Eisenstein, 2008).

We consider four routing methods, each of which is indicated by a single letter in the formulas: r(return routing), s(S shape), g(largest gap) *or* m(midpoint). We let ρ be a placeholder in formulas for the routing method if results hold for all routing methods. D^ρ denotes the expected route length of an order for a given storage location assignment A, after correcting for empty orders. Note that strictly speaking $D^\rho(A)$ would be a more correct way of notation, however, for ease of notation the evident dependency on A is omitted in the notation; this also applies to several other variables. We let L_c and L_i^ρ denote respectively the distance traveled in the two cross aisles and in aisle i for any order picking route and a given storage location assignment A, before correcting for empty orders. Let $\mathbb{E}(L_c)$ and $\mathbb{E}(L_i^\rho)$ denote their respective expected values.

We let P_i denote the probability of having at least one pick item in aisle i, that is $P_i = 1 - \prod_{j=1}^{n}(1 - p_{ij})$. And let P_i^ℓ be the probability of no pick items in aisles $i+1, \ldots, m$ (ℓ of "last") and P_i^f the probability of no pick items occurring in aisles $1, \ldots, i-1$ (f of "first"). Thus $P_i^f = \prod_{j=1}^{i-1}(1 - P_j)$ and $P_i^\ell = \prod_{j=i+1}^{m}(1 - P_j)$.

Below we summarize the main variables and parameters as used in the remainder of this paper. As the expected route length depends on the routing method chosen, in the next section we first derive exact analytical formulas for the objective function for four routing methods. Following that, we present a method for finding storage location assignments.

For the layout, we have the following parameters (see also Fig. 1):

m	Number of aisles
n	Number of storage locations per aisle
w_a	Width of an aisle
w_c	Distance from middle of a cross aisle to the head of an aisle
f	Distance between two adjacent storage locations
i	$\in \{1, \ldots, m\}$. An index for the aisles
j	$\in \{1, \ldots, n\}$. An index for the locations in an aisle

To describe the orders, we have the following parameters:

K	The set of all items
k	$\in K$. An index to indicate an item
p_k	Probability that item k is in an order
$\mathscr{P}$	Probability of having an order of positive length
X_k	Binary variable indicating whether ($X_k = 1$) or not ($X_k = 0$) item k is on an order
O	A specific order. Collection of all k for which $X_k = 1$

For the route length calculations we have the variables:

$\mathbb{E}(\mathbb{L}_c)$	Expected distance traveled in the two cross aisles, not corrected for empty orders
$\mathbb{E}(\mathbb{L}_i^\rho)$	Expected distance traveled in aisle i for routing method ρ, not corrected for empty orders
$\mathbb{E}(\mathbb{L}^\rho)$	Expected distance traveled in aisles $1, \ldots, m$ for routing method ρ, not corrected for empty orders
D^ρ	Expected total route length of orders for routing method ρ
p_{ij}	Probability of the item assigned to storage location j in aisle i being in an order
P_i	Probability of having to enter aisle i
P_i^ℓ	Probability of having no pick items in aisles $i+1, \ldots, m$
P_i^f	Probability of having no pick items in aisles $1, \ldots, i-1$

6. Exact formulas for average route length

In this section, we present the analytical formulas for the average route length for the four routing methods, return, S-shape, midpoint and largest gap. First, we present formulas for the expected travel distances within one aisle for each routing method, not conditioned on the absence of empty orders. After that, we present a formula for the travel distance in the cross aisles, that holds for all four routing methods, as well as the total expected route length formula. We have chosen this organization since the dynamic program for storage assignment in Section 8 relies on the formulas for travel distance within the individual aisles.

6.1. Return routing

We determine the distance for traveling in aisle i with the return routing method, empty orders not excluded. In Section 6.5, we add the formula for distances traveled in the cross aisles and rule out the empty orders to create the total formula for route length. The distance an order picker has to travel in an aisle for return routing is solely determined by whether there is a pick item in that aisle and if so, which pick item is furthest from the front cross aisle.

For an arbitrary order, the distance traveled in aisle $i, i = 1, \ldots m$, equals:

$$L_i^r = \begin{cases} 0 & \text{if no product has to be picked in aisle } i, \\ 2(w_c + jf - \frac{1}{2}f) & \text{if the product located at } j \text{ is the furthest product to be picked in aisle } i. \end{cases}$$

This can be explained as follows. Starting in the middle of the front cross aisle, the picker travels to the head of the aisle (w_c). Then he passes along $j-1$ locations, each of length f. He may stop at some of these locations to perform a pick. Subsequently, the picker has reached the last location that must be visited, and travel to the middle of it ($\frac{1}{2}f$). To return to the front cross aisle, the same distance must be traveled again. The expected travel distance can be found by multiplying the distance corresponding to location j by the probability of the product located at that location being the furthest product to be picked, for all locations j and aisles i.

Using conditioning, we obtain as expected value:

$$\begin{aligned} \mathbb{E}(L_i^r) &= 2\sum_{j=1}^{n}\left(\left(w_c + jf - \frac{1}{2}f\right)p_{ij}\prod_{h=j+1}^{n}(1-p_{ih})\right) \\ &= 2P_i w_c + 2\sum_{j=1}^{n}\left(\left(jf - \frac{1}{2}f\right)p_{ij}\prod_{h=j+1}^{n}(1-p_{ih})\right). \end{aligned} \tag{1}$$

The last equality follows from the property

$$\sum_{j=1}^{n} p_{ij}\prod_{h=j+1}^{n}(1-p_{ih}) = 1 - \prod_{j=1}^{n}(1-p_{ij}),$$

which can be proven by induction.

6.2. S-shape routing

For S-shape routing we present here, as with return routing, the formula for the distances traveled within the aisles, without removing empty orders. We start by studying the last aisle that needs to be visited. If the number of aisles to visit is even, then the total distance in all aisles is obtained simply by multiplying the expected number of visited aisles with the length of an aisle. However, if the number of aisles to visit is odd then in the last aisle a turn must be made, which causes the travel distance in the last aisle to be identical to return routing, see Fig. 2. First we introduce one additional variable, P_i^e, the probability that an even number of aisles must be visited before aisle i (i.e., in aisles $1, \ldots, i-1$). For P_i^e the following recursive relation holds:

$$P_i^e = P_{i-1}^e(1 - P_{i-1}) + (1 - P_{i-1}^e)P_{i-1},$$

with $P_1^e \equiv 1$. The expected travel distance in aisle i for S-shape routing can now be written as:

$$\mathbb{E}(L_i^s) = (1 - P_i^e P_i^\ell)P_i(2w_c + nf) + P_i^e P_i^\ell \mathbb{E}(L_i^r).$$

The formula can be explained as follows. The term $P_i^e P_i^\ell$ gives the probability that in aisle i a turn must be made; that is, the probability that the number of aisles visited before aisle i is even and no aisles need to be visited after aisle i. The distance to be traveled in the aisle then equals the length of the return routing method, i.e., $\mathbb{E}(L_i^r)$. The probability of not having to make a turn in aisle i is $(1 - P_i^e P_i^\ell)$ in which case the length of an aisle $(2w_c + nf)$ must be added if the aisle has pick items, i.e., with probability P_i.

6.3. Largest-gap routing

We first define G_i to be the expected largest gap in aisle i. We use a recursive approach to determine G_i. Let the largest gap observed up to location j be g_ℓ, conditioning on the realization of the picks in aisle i up to and including location j. Furthermore, let g_c be the gap that is currently observed based on the realization of picks; this is the distance from the pick closest to location j to location $j+1$. Finally, define $G_{ij}(g_c, g_\ell)$ to be the expected largest gap in aisle i, given a g_c and g_ℓ corresponding to location j. Starting from the end of the aisle we recursively go over all locations in the aisle to determine $G_{ij}(g_c, g_\ell)$.

The expected largest gap can be found by solving the following set of recursive equations:

$$
\begin{aligned}
&G_i \equiv G_{i0}\left(\frac{1}{2}f + w_c, \frac{1}{2}f + w_c\right)\\
&G_{i,n+1}(g_c, g_\ell) \equiv g_\ell\\
&G_{in}(g_c, g_\ell) = (1-p_{in})G_{i,n+1}\left(g_c + \frac{1}{2}f + w_c, \max\left\{g_\ell; g_c + \frac{1}{2}f + w_c\right\}\right)\\
&\qquad\qquad + p_{in}G_{i,n+1}\left(\frac{1}{2}f + w_c, \max\left\{g_\ell; \frac{1}{2}f + w_c\right\}\right)\\
&G_{ij}(g_c, g_\ell) = (1-p_{ij})G_{i,j+1}(g_c + f, \max\{g_\ell; g_c + f\}) + p_{ij}G_{i,j+1}(f, \max\{g_\ell; f\}).
\end{aligned}
$$

For further details see Appendix B. The expected travel length in aisle i for largest-gap routing can then be written as:

$$\mathbb{E}(L_i^g) = P_i^\ell P_i^f \mathbb{E}(L_i^r) + (P_i^\ell + P_i^f - 2P_i^\ell P_i^f)P_i(2w_c + nf) + (1-P_i^\ell)(1-P_i^f)2(2w_c + nf - G_i).$$

This formula can be explained as follows. The first term accounts for situations when aisle i is the only aisle with pick items, which has a probability of $P_i^\ell P_i^f$ and a travel distance identical to return routing in one aisle, $\mathbb{E}(L_i^r)$. The second term accounts for situations in which aisle i is either the first or last aisle to visit (but not both), i.e., is traversed entirely with travel distance $2w_c + nf$. The third term account for situations in which aisle i is neither the first nor the last aisle, and hence entered and left from both sides with a distance equal to two times the aisle length minus two times the largest gap, $2(2w_c + nf) - 2G_i$.

6.4. Midpoint routing

The formula for the midpoint routing method is quite similar to that for the largest-gap routing. Only the turns in the aisles are made such that the gap always crosses the middle of the aisle. Or put differently, the distance traveled in such an aisle is comparable to performing return routing twice on both halves of the aisle. Letting $i(1)$ and $i(2)$ refer to the upper and lower half of aisle i respectively, we obtain, similar to the distance estimate for return routing (Eq. (1)):

$$
\begin{aligned}
\mathbb{E}(L_{i(1)}^m) &= 2\sum_{j=\lceil n/2\rceil+1}^{n}\left(\left(w_c + (n-j)f + \frac{1}{2}f\right)p_{ij}\prod_{h=\lceil n/2\rceil+1}^{j-1}(1-p_{ih})\right)\\
\mathbb{E}(L_{i(2)}^m) &= 2\sum_{j=1}^{\lceil n/2\rceil}\left(\left(w_c + jf - \frac{1}{2}f\right)p_{ij}\prod_{h=j+1}^{\lceil n/2\rceil}(1-p_{ih})\right)
\end{aligned}
$$

Very similar to the formula for largest-gap routing, we then have for the expected travel length in aisle i for midpoint routing:

$$\mathbb{E}(L_i^m) = P_i^\ell P_i^f \mathbb{E}(L_i^r) + (P_i^\ell + P_i^f - 2P_i^\ell P_i^f)P_i(2w_c + nf) + (1-P_i^\ell)(1-P_i^f)(\mathbb{E}(L_{i(1)}^m) + \mathbb{E}(L_{i(2)}^m)).$$

6.5. Cross aisle distances and total route length

As was already noted in Roodbergen and Vis (2006), cross aisle distances do not differ between the S-shape routing method and the largest-gap routing method. The same reasoning applies to return routing and midpoint routing. Hence it suffices to have one formula for the expected travel distance in the cross aisle.

The expected incremental value added to cross aisle distances by aisle i, is given by:

$$\mathbb{E}(L_i^c) = 2w_a(i-1)P_iP_i^\ell.$$

That is, if we have to visit aisle i and none of the aisles $i+1,\ldots,m$, which has a probability $P_iP_i^\ell$ of occurring, then we incur a travel distance of $2w_a(i-1)$. Total route length for routing method $\rho(=r,s,g,\text{ or } m)$ now can be obtained as:

$$D^\rho = \frac{1}{\mathscr{P}}\left(\sum_{i=1}^{m}\mathbb{E}(L_i^\rho) + \mathbb{E}(L_i^c)\right),$$

which is in this form also conditioned on the absence of empty orders. The formulas derived above remain valid for situations in the case the storage locations on both sides of an aisle are not merged into one column. Deriving the expected travel distance in case two products $\hat{k}$ and $\bar{k}$ are stored opposite each other can be done by using a dummy product k with $p_k = 1 - (1 - p_{\hat{k}})(1 - p_{\bar{k}})$. Additionally, in case of nonidentical storage space, using the appropriate distance function for location j instead of the linear $w_c + jf - \frac{1}{2}f$ will lead to the expected travel distance function. However, this may result in exponential calculation times for largest-gap routing.

7. Optimality conditions

The mathematical model defined in the previous sections is non-linear and non-convex and hence is in general hard to solve. In this section we prove some properties of optimal solutions that will aid later in our solution procedures.

We show that for any optimal storage location assignment under return routing, the items are stored in an ordering based on their respective picking probabilities, both across aisles and inside the aisles.

Theorem 1. *Given that $p_k < 1,\ \forall k \in K$, a necessary condition for a solution to be optimal for the storage location assignment problem under return routing is:*

$$p_{i1} \geqslant p_{i2} \geqslant \cdots \geqslant p_{in} \quad \forall i \in \{1, \ldots, m\}, \tag{2}$$

$$p_{1j} \geqslant p_{2j} \geqslant \cdots \geqslant p_{mj} \quad \forall j \in \{1, \ldots, n\}. \tag{3}$$

Proof. See Appendix A. □

Theorem 1 extends results that have been found for one-dimensional settings, i.e., settings with only a single aisle (Eisenstein, 2008; Chou et al., 2012). It is interesting to consider class-based storage for return routing in the context of Theorem 1. Since, according to the theorem, the items have to be sorted both vertically (within the aisle) and horizontally (across the aisles), it follows that the class boundaries must be non-increasing functions when plotting the highest location number of each class as a function of the aisle number. For example, looking at Fig. 3, the storage location assignment on the left complies with Theorem 1 and may therefore be optimal. The storage location assignment on the right, however, does not comply with the theorem and is therefore known not to be optimal, without the need of calculating route length. This fact can easily be seen in the figure when comparing the positions of the A-items in aisles 2 and 3. This insight drastically decreases the number of solutions that need to be evaluated in an exhaustive search.

Under S-shape routing, there are only two options for visiting an aisle. The common way of visiting an aisle is to traverse it completely. If, however, an odd number of aisles must be visited, a turn is made in the last aisle. This turn is identical to the turn that would have been made under return routing. For this turn in the last aisle, we know Eq. (2) of Theorem 1 holds. And since all other aisles are indifferent to the aisle's storage location assignment (because we either do not traverse it or traverse it entirely), it follows that Eq. (2) also holds for S-shape routing. This gives us Corollary 2.

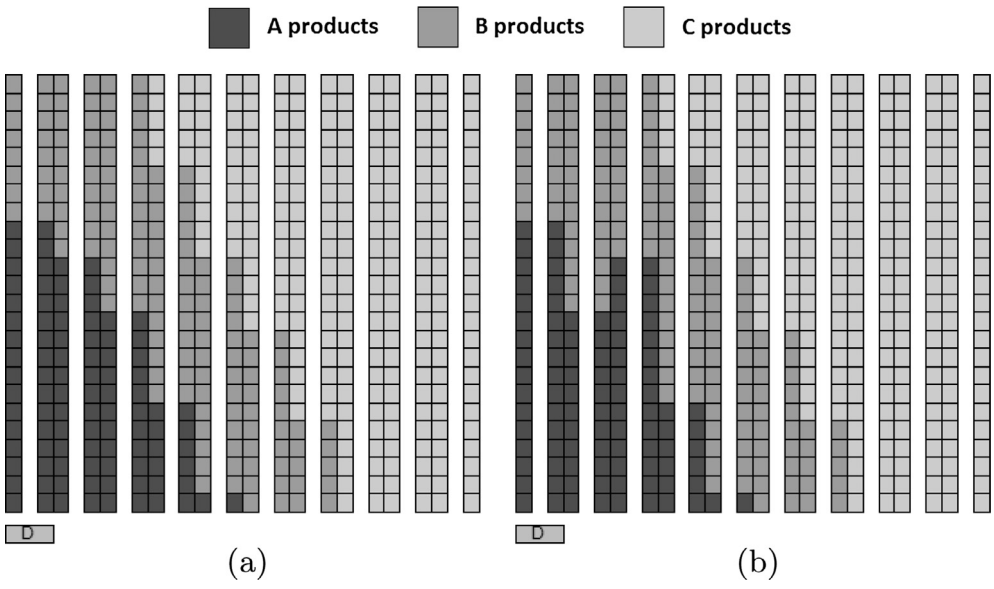


Fig. 3. Class based storage with multiple classes. The left one might be optimal for return routing, the right cannot be optimal.

Corollary 2. *Given that $p_k < 1,\ \forall k \in K$, a necessary condition for a solution to be optimal for the storage location assignment problem under S-shape routing is:*

$$p_{i1} \geqslant p_{i2} \geqslant \cdots \geqslant p_{in} \quad \forall i \in \{1, \ldots, m\}.$$

A similar remark can be made for largest-gap and midpoint routing. When there is a pick item in aisle 1 or in aisle m, then we know that a route based on the largest-gap or midpoint routing methods traverses this aisle entirely, or, if this is the only aisle to visit, a return is made. Thus, similar to S-shape routing it is either traversed completely or picked by a return route. Hence, very similar to Corollary 2 for S-shape, we have the following.

Corollary 3. *Given that $p_k < 1,\ \forall k \in K$, a necessary condition for a solution to be optimal for the storage location assignment problem under largest-gap and midpoint routing is:*

$$p_{i1} \geqslant p_{i2} \geqslant \cdots \geqslant p_{in} \quad i \in \{1, m\}.$$

8. Solution method

The storage location assignment problem is a problem with a typically very large solution space and a complex objective function. This may likely be the reason why only few articles on storage location assignment provide systematic procedures for finding good or optimal solutions for the problem. In Section 6, we have presented new formulas for determining the exact average route length that can be determined with low computational effort, which provides a first step in making the problem more tractable. In Section 7, we derived some conditions that reduce the solution space that needs to be evaluated, which provides a second step in making the problem more tractable. In this Section, we give a structured solution procedure, based on Dynamic Programming, that incorporates the results from Sections 6 and 7.

We first describe the dynamic program. Assume class-based storage with Q classes. Note that this does not exclude the possibility for each item to be treated individually (i.e., $Q = |K|$), though it is unlikely that the presented approach would then be tractable for realistically sized problems. Let the number of items in each class q be denoted by a parameter x_q and the probability of an item from class q to be in an order p_q, for all $q \in Q$. The parameter vector x is defined as $x = (x_1, x_2, \ldots, x_Q)$. States are described by means of the vector $y_i = (y_1^i, y_2^i, \ldots, y_Q^i)$, which basically is a tally for the number of items of each class that have been used so far. The set of feasible states at the stage corresponding to aisle i is given by $Y_i = \left\{ y_i \middle| \sum_{q=1}^{Q} y_q^i = n \times i, \text{ and } y_q^i \leqslant x_q \text{ for all } q \in Q \right\}$. That is, from the available items, we must have selected exactly an amount that fit into i aisles. The size of the state space of aisle i is polynomial in the size of the warehouse, as $|Y_i| = \mathcal{O}((mn)^{Q-1})$. The forward dynamic program starts from state $y_0 \equiv 0$. The first transition adds aisle 1. We evaluate every possible state $y_1 \in Y_1$. That is, we select a total of n items from all Q classes together, and since there is only one aisle under consideration, assign these to aisle 1. We determine the best assignment to locations within aisle 1 for these items, and determine the corresponding costs.

To make a transition from aisle $i-1$ to aisle i, we evaluate every possible state $y_i \in Y_i$. That is, we select a total of $n \times i$ items from all Q classes together. Costs for state y_i are obtained by finding a feasible cost minimizing combination of a state y_{i-1}, as previously determined in stage $i-1$, an assignment $z_i \equiv y_i - y_{i-1}$ of items to aisle i, and an assignment of items to specific locations in aisle i. Only states can be used for which $z_i \geqslant 0$. Since z_i is the allocation of products to a single aisle and all aisles are identical, we have $z_i \in Y_1$.

The last transition involves aisle m. For each state y_{m-1}, as previously determined the stage corresponding to aisle $m-1$, we assign $x - y_{m-1}$ items to aisle m, find the best assignment of these items in aisle m, and choose the best of these options as our final solution.

Note that assigning a given set of m items to appropriate locations within an aisle may by itself already be a non-trivial task. Furthermore, the above structure implies that the best way to assign items to locations in aisles $1, \ldots, i$, does not depend on later decisions for assigning items to locations in aisles $i+1, \ldots, m$. This is not always true, in which case the dynamic program serves as a heuristic instead of an optimal algorithm. These specifics are discussed below, when introducing the transition costs $c_i^{\rho}(y_i, z_i)$ for the respective routing methods.

The value functions can now be written as:

$$V_i(y_i) = \min_{\substack{z_i \in Y_1 \\ y_i - y_{i-1} = z_i}} \left(V_{i-1}(y_{i-1}) + c_i^{\rho}(y_i, z_i) \right) \quad \text{for } i = 1, \ldots, m$$

$$V_0(y_0) = 0.$$

8.1. Transition costs

For the transition costs in the dynamic program, we use:

$$c_i^{\rho}(y_i, z_i) = \frac{1}{\mathscr{P}}\left(\mathbb{E}(L_i^{\rho}) + \mathbb{E}(L_i^{c})\right).$$

Note that $\mathbb{E}(L_i^{\rho})$ and $\mathbb{E}(L_i^{c})$ can only be determined for a given storage location assignment $a_i = (a_{i1}, \ldots, a_{in})$. In our dynamic program, however, we only specify (through z_i) which items are to be placed in aisle i but not at which locations within the aisle. For each of the four routing methods, we discuss below the approach to obtain a specific storage location assignment within the aisle for a given z_i, and some other relevant considerations.

For return routing, due to Theorem 1, we can easily determine an optimal permutation $\pi(z_i)$ of z_i in aisle i. Therefore, transition costs for return routing can be determined quickly. This allows us to obtain optimal solutions in low calculation times. The only remaining aspect that may need some attention in the calculation of $c_i^{r}(y_i, z_i)$ is P_i^{ℓ}, since this depends on the location assignment in aisles that have not yet been evaluated by the dynamic program. However, note that an alternative way of calculating P_i^{ℓ} is given by:

$$P_i^{\ell} = \prod_{t=1}^{T}(1 - p_t)^{x_t - y_t^i}.$$

For S-shape routing, we can for any given z_i determine an optimal permutation $\pi(z_i)$, due to Corollary 2. However, a requirement for the dynamic program in this setup to yield optimal solutions, is that all components of $c_i^{s}(y_i, z_i)$ depend only on y_i and z_i. This is not the case. It holds for all factors in the expression, except for P_i^{e}, which is the probability that an even number of aisles must be visited before aisle i. To determine $\widehat{P}_i^{e}$, we need not only know y_i, but also the exact division of items among aisle $1, \ldots, i-1$. After all, different divisions lead to alterations in the probabilities of having to enter the aisles, which are underlying in the calculation for determining the probability of visiting an even number of aisles. This information is not retained in the setup of our dynamic program, and including this information would result in an intractably large state space.

This implies that we have no guarantee that the dynamic program produces optimal solutions with the presented transition cost formula for S-shape routing. However, we were unable to think of any systematic biases that may result. The main driver for improving storage location assignment under S-shape routing is to minimize the number of aisles to visit, which does not depend on P_i^{e}. By complete enumeration, we were able to obtain optimal solutions under S-shape routing, which was feasible due to solution space restrictions following from the optimality conditions derived in Section 7. Comparisons between optimal solutions and solutions from the dynamic program show that the difference is insignificant. These results are presented in Section 9.

For largest-gap routing, we do not have any optimality conditions that allows us to easily convert z_i into an optimal storage location assignment within an aisle. We therefore limit the storage location assignment within aisles to two specific permutations. Order z such that $z_{(1)} \geqslant z_{(2)} \geqslant z_{(3)} \geqslant \cdots \geqslant z_{(n)}$. We consider the permutations $(z_{(1)}, z_{(2)}, \ldots, z_{(n)})$ and $(z_{(1)}, z_{(3)}, \ldots, z_{(n-1)}, z_{(n)}, \ldots, p_{(4)}, p_{(2)})$, which order the products from highest picking probability to lowest probability and symmetric in the aisle respectively. We allow both permutations, and make the choice as integral part of the dynamic program. Note that the combination of these two permutations is quite versatile, and can induce a wide variety of storage shapes, including all a priori shapes from literature, such as diagonal storage, perimeter storage, within-aisle storage and across-aisle storage. We use the same permutations for midpoint routing. As a consequence, solutions for storage location assignment under largest-gap and midpoint routing need not be optimal, i.e., the dynamic program serves as a heuristic.

Finding the best assignment takes the evaluation of $\mathcal{O}(n^{Q-1})$ different assignments. The evaluation of an assignment of items to an aisle, as described above, can make use of precomputed values for expected largest-gap distance as well as expected return distance. This implies that the evaluation of an assignment of items to an aisle can be done in constant time. The size of the state space is of order $\mathcal{O}((mn)^{Q-1})$ for each aisle. Finally, the value functions need to be computed for m aisles. Hence, computing the value functions of the DP has complexity $\mathcal{O}(m) \cdot \mathcal{O}(n^{Q-1}) \cdot \mathcal{O}((mn)^{Q-1}) = \mathcal{O}(m^{Q} n^{2(Q-1)})$.

9. Numerical experiments

We have implemented the formulas for expected route length and the dynamic programming method in Julia. We study a picking area with $w_a = 2,\ w_c = 0.5,\ f = 1,\ n = 24$ employing class-based storage with three classes (A, B, and C). The settings we use in these experiments are composed as follows. The picking area has either 7 or 15 aisles. The probabilities for items of each class to be in an order are set such that the average number of pick items per route is either 2, 10 or 20 and the probability of a pick item coming from classes A/B/C is either 80/15/5 percent or 50/30/20 percent. The classes A, B, and C span respectively 20%, 30% and 50% of the available storage space. This results in 12 sets of probabilities (p_A, p_B, p_C) for an item of a class to be in an order, namely $\left(\frac{4}{85}, \frac{3}{500}, \frac{1}{840}\right)$, $\left(\frac{1}{34}, \frac{3}{250}, \frac{1}{210}\right)$, $\left(\frac{4}{17}, \frac{3}{100}, \frac{1}{168}\right)$, $\left(\frac{5}{34}, \frac{3}{50}, \frac{1}{42}\right)$, $\left(\frac{8}{17}, \frac{3}{50}, \frac{1}{84}\right)$, $\left(\frac{5}{17}, \frac{3}{25}, \frac{1}{21}\right)$, $\left(\frac{1}{45}, \frac{1}{360}, \frac{1}{1800}\right)$, $\left(\frac{1}{72}, \frac{1}{180}, \frac{1}{450}\right)$, $\left(\frac{1}{9}, \frac{1}{72}, \frac{1}{360}\right)$, $\left(\frac{5}{72}, \frac{1}{36}, \frac{1}{90}\right)$, $\left(\frac{2}{9}, \frac{1}{36}, \frac{1}{180}\right)$, and $\left(\frac{5}{36}, \frac{1}{18}, \frac{1}{45}\right)$. For each setting and for each routing method, we determine the average route length under four common storage location assignment rules: across-aisle storage (AA), within-aisle storage (WA), diagonal storage (DI), and perimeter (PE) storage (cf. Petersen and Schmenner, 1999). We compare these to the storage location assignment obtained by our dynamic programming algorithm (DP). Calculation times are reported for a 3.30 GHz Intel i3-3220 CPU.

Table 1
Expected route lengths for a number of settings under return routing. Bold indicates the minimum value of the columns AA, WA, DI and PE.

Aisles	Picks per route	Pick percentage	AA	WA	DI	PE	Our DP	CPU (s)
7	2	80/15/5	35.88	44.77	**34.69**	53.98	34.30	4.69
7	2	50/30/20	49.83	57.80	**49.58**	63.75	49.14	4.67
7	10	80/15/5	**89.97**	106.70	91.30	121.49	89.56	4.69
7	10	50/30/20	**134.94**	156.94	137.64	167.06	134.89	4.71
7	20	80/15/5	**128.08**	145.77	128.24	163.30	127.34	4.70
7	20	50/30/20	**195.37**	219.54	198.04	234.57	195.37	4.71
15	2	80/15/5	57.12	63.49	**50.75**	91.13	49.92	58.87
15	2	50/30/20	71.96	78.51	**70.04**	92.92	68.79	59.00
15	10	80/15/5	**130.03**	172.01	140.02	225.77	128.40	59.14
15	10	50/30/20	**184.28**	227.56	198.72	249.77	183.89	59.87
15	20	80/15/5	**190.91**	242.84	205.88	334.23	190.29	62.19
15	20	50/30/20	**283.03**	342.33	304.58	380.78	282.91	59.82

Table 2
Expected route lengths for a number of settings under S-shape routing. Bold indicates the minimum value of the columns AA, WA, DI and PE.

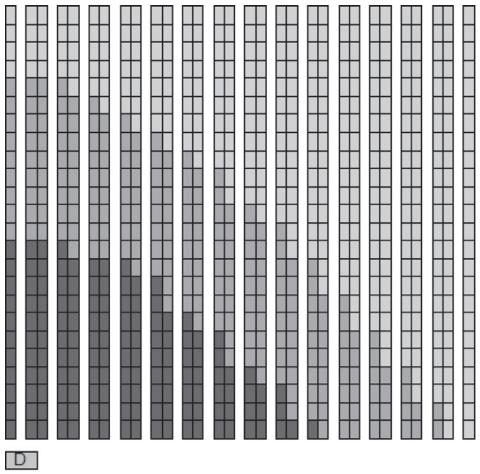
Aisles	Picks per route	Pick percentage	AA	WA	DI	PE	Our DP	CPU (s)
7	2	80/15/5	58.19	**46.42**	51.28	54.72	45.89	9.06
7	2	50/30/20	62.00	**58.06**	59.46	63.17	57.76	11.14
7	10	80/15/5	155.25	**93.18**	129.59	105.51	92.82	7.50
7	10	50/30/20	156.67	**136.34**	150.29	143.99	136.28	7.47
7	20	80/15/5	185.60	**119.77**	159.93	135.45	119.69	7.50
7	20	50/30/20	192.03	**172.08**	185.78	182.56	172.01	7.60
15	2	80/15/5	82.17	**61.37**	64.63	90.28	60.88	115.07
15	2	50/30/20	86.27	**77.50**	78.15	92.13	77.14	104.23
15	10	80/15/5	232.59	**144.39**	180.27	207.70	144.33	94.66
15	10	50/30/20	233.65	**206.11**	217.30	230.17	206.11	94.42
15	20	80/15/5	336.67	**191.44**	246.32	288.65	191.44	94.24
15	20	50/30/20	334.62	**287.45**	306.76	322.85	287.45	94.64

Table 3
Expected route lengths for a number of settings under largest-gap routing. Bold indicates the minimum value of the columns AA, WA, DI and PE.

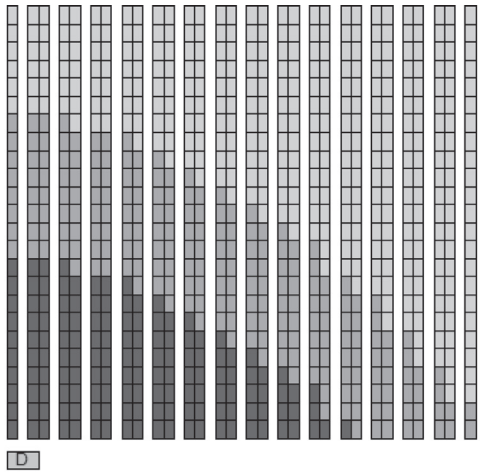
Aisles	Picks per route	Pick percentage	AA	WA	DI	PE	Our DP	CPU (s)
7	2	80/15/5	55.12	**45.16**	49.44	53.43	45.07	29.40
7	2	50/30/20	58.70	**54.92**	56.22	59.41	54.58	26.47
7	10	80/15/5	110.69	**83.71**	103.04	85.74	79.98	26.35
7	10	50/30/20	124.95	116.74	122.94	**109.68**	108.49	26.53
7	20	80/15/5	138.60	108.79	129.08	**98.32**	98.32	26.37
7	20	50/30/20	166.02	157.69	164.12	**140.47**	140.47	26.30
15	2	80/15/5	77.36	**58.38**	61.94	82.78	58.35	170.09
15	2	50/30/20	81.40	**72.58**	73.48	84.68	72.10	171.67
15	10	80/15/5	152.01	130.06	141.90	**128.42**	118.44	172.75
15	10	50/30/20	170.80	167.09	168.73	**156.37**	151.43	172.52
15	20	80/15/5	203.34	180.87	193.78	**152.24**	152.01	183.98
15	20	50/30/20	242.18	241.05	242.96	**209.27**	209.25	187.35

Table 4
Expected route lengths for a number of settings under midpoint routing. Bold indicates the minimum value of the columns AA, WA, DI and PE.

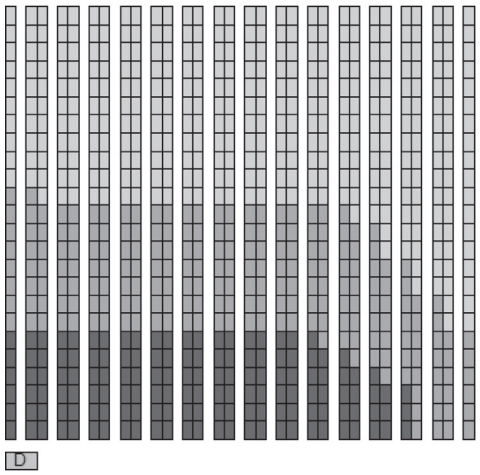
Aisles	Picks per route	Pick percentage	AA	WA	DI	PE	Our DP	CPU (s)
7	2	80/15/5	55.12	**45.18**	49.46	53.43	45.10	12.64
7	2	50/30/20	58.73	**54.98**	56.28	59.42	54.63	12.60
7	10	80/15/5	111.03	**84.55**	104.19	85.82	80.21	12.67
7	10	50/30/20	126.85	119.41	125.67	**110.64**	109.40	12.74
7	20	80/15/5	139.77	111.50	132.49	**98.62**	98.62	12.67
7	20	50/30/20	171.84	164.93	172.08	**143.57**	143.57	12.66
15	2	80/15/5	77.37	**58.48**	61.99	82.78	58.43	152.85
15	2	50/30/20	81.43	**72.64**	73.53	84.69	72.15	156.18
15	10	80/15/5	152.22	134.19	144.51	**128.48**	118.57	158.70
15	10	50/30/20	172.07	170.45	171.47	**157.02**	152.07	158.51
15	20	80/15/5	204.17	189.64	201.87	**152.46**	152.46	157.14
15	20	50/30/20	246.89	251.05	252.18	**211.64**	211.63	151.32



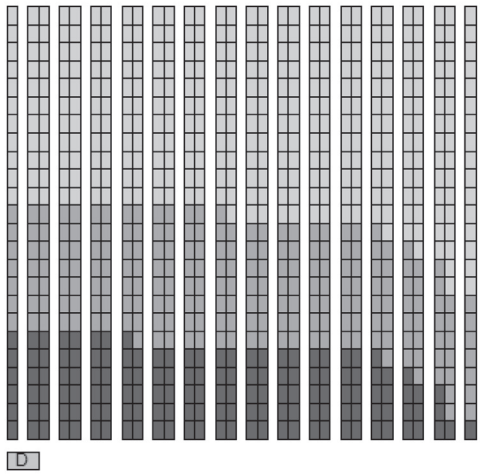
(a) Order length 2, distribution 80, 15, 5.



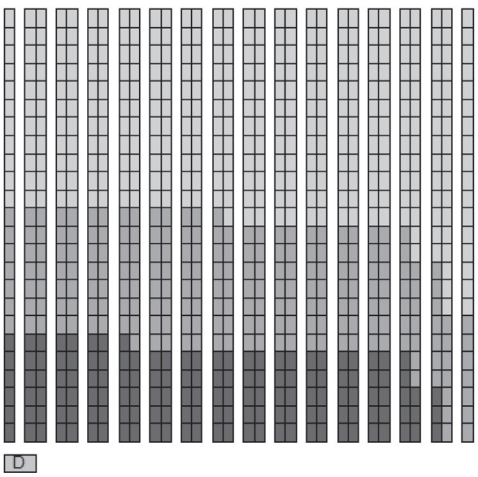
(b) Order length 2, distribution 50, 30, 20.



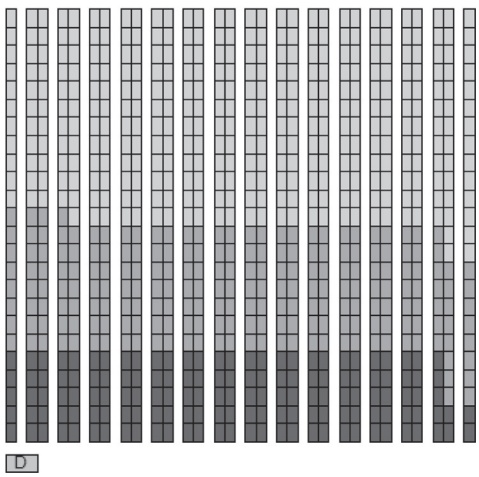
(c) Order length 10, distribution 80, 15, 5.



(d) Order length 10, distribution 50, 30, 20.

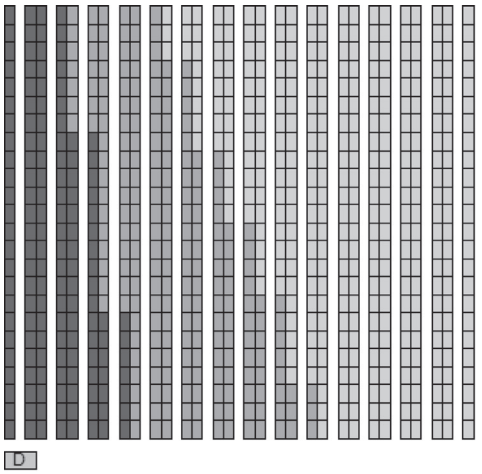


(e) Order length 20, distribution 80, 15, 5.

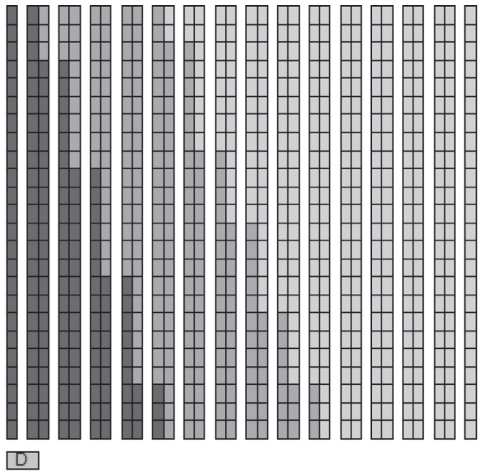


(f) Order length 20, distribution 50, 30, 20.

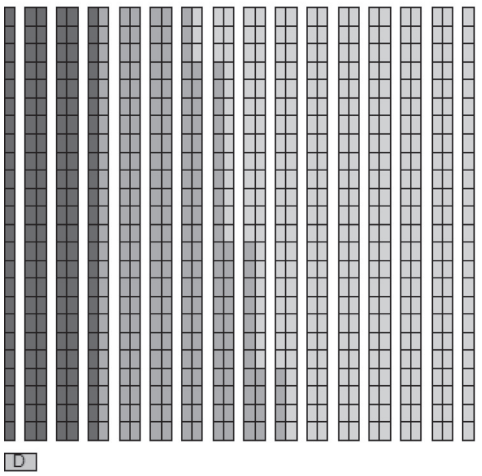
Fig. 4. Optimal storage location assignments under return routing for different expected order lengths and demand distributions.



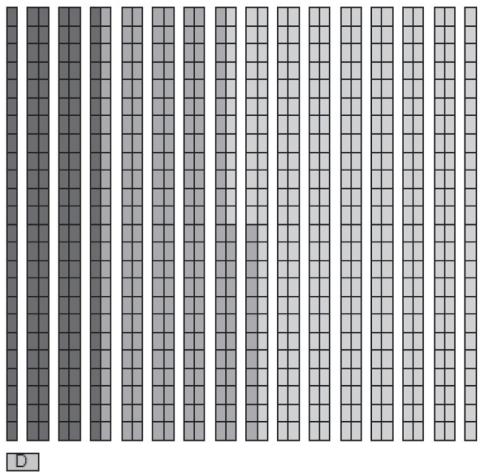
(a) Order length 2, distribution 80, 15, 5.



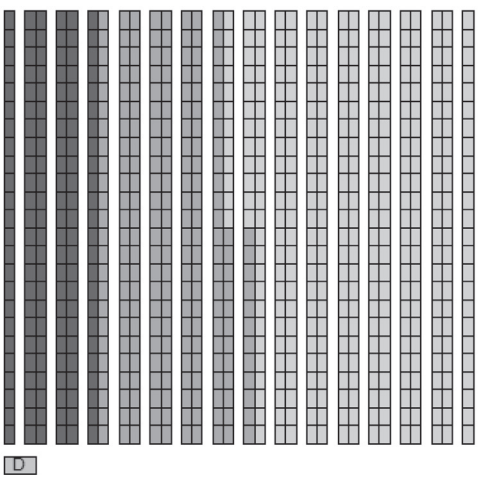
(b) Order length 2, distribution 50, 30, 20.



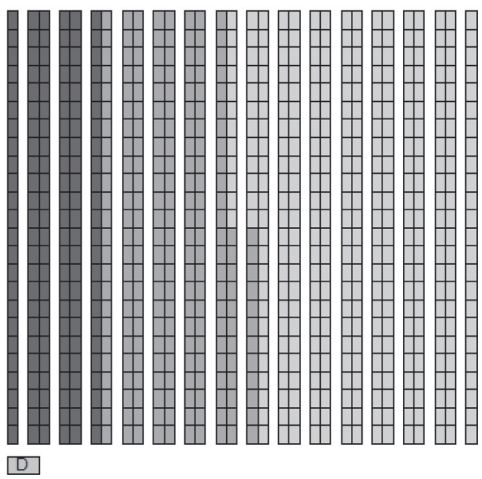
(c) Order length 10, distribution 80, 15, 5.



(d) Order length 10, distribution 50, 30, 20.

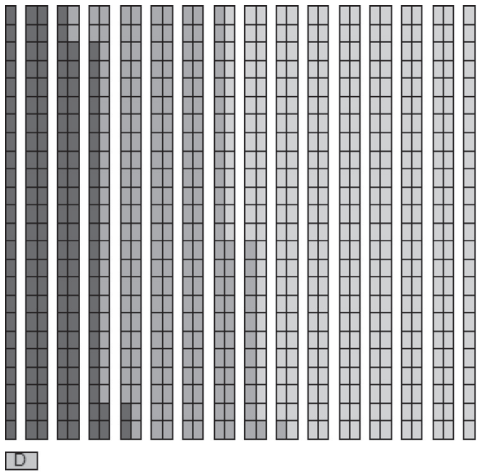


(e) Order length 20, distribution 80, 15, 5.

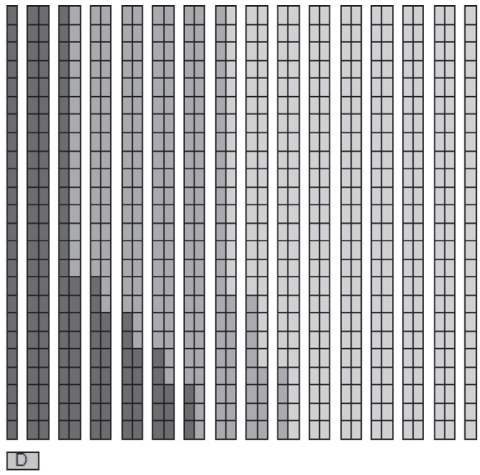


(f) Order length 20, distribution 50, 30, 20.

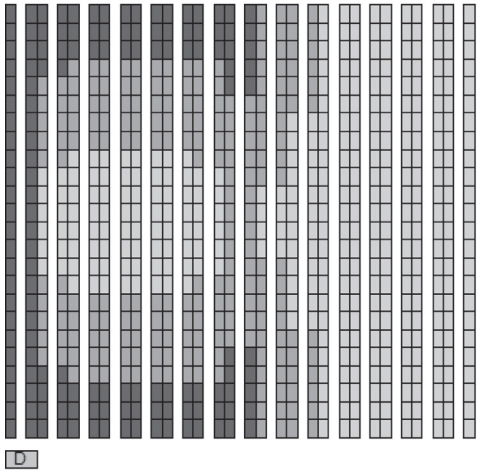
Fig. 5. Storage location assignments for S-shape routing produced by the DP for different expected order lengths and demand distributions.



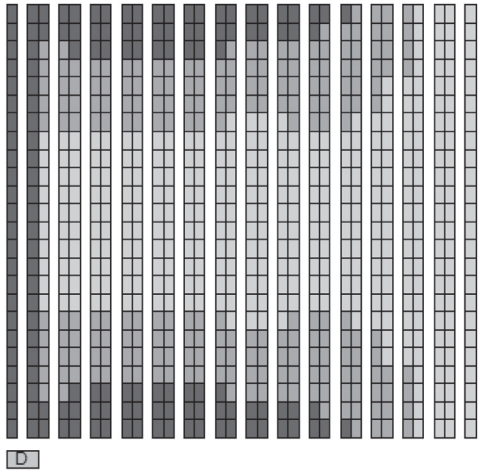
(a) Order length 2, distribution 80, 15, 5.



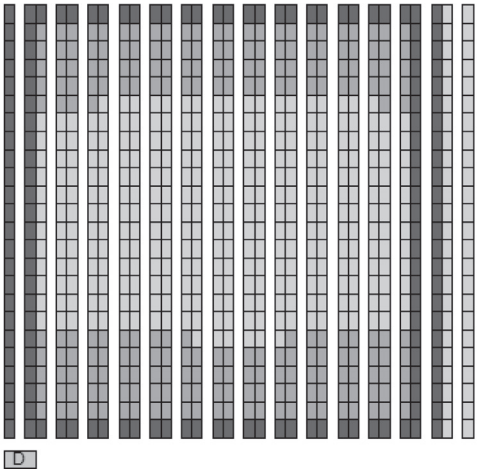
(b) Order length 2, distribution 50, 30, 20.



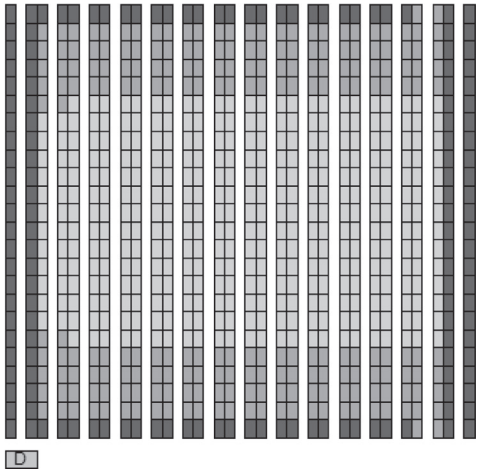
(c) Order length 10, distribution 80, 15, 5.



(d) Order length 10, distribution 50, 30, 20.

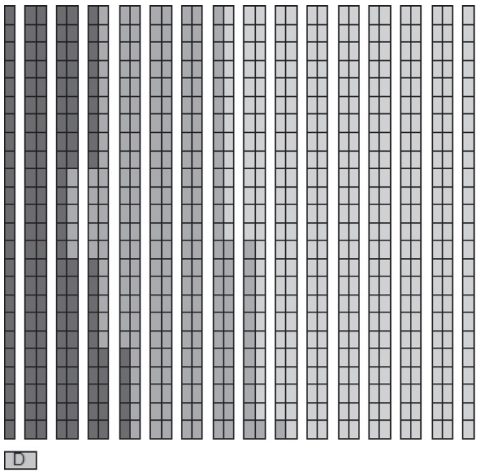


(e) Order length 20, distribution 80, 15, 5.

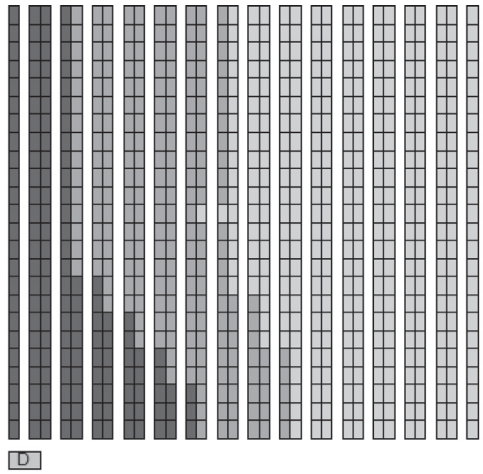


(f) Order length 20, distribution 50, 30, 20.

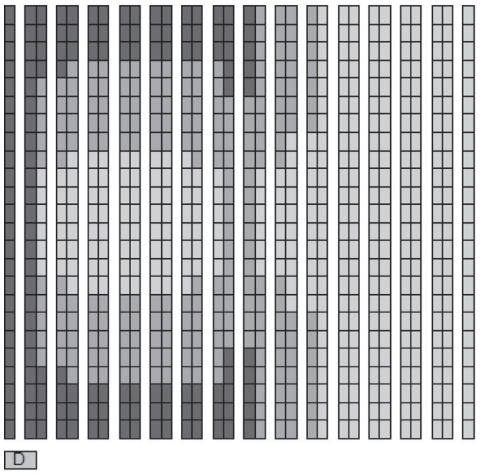
Fig. 6. Storage location assignments for largest-gap routing produced by the DP for different expected order lengths and demand distributions.



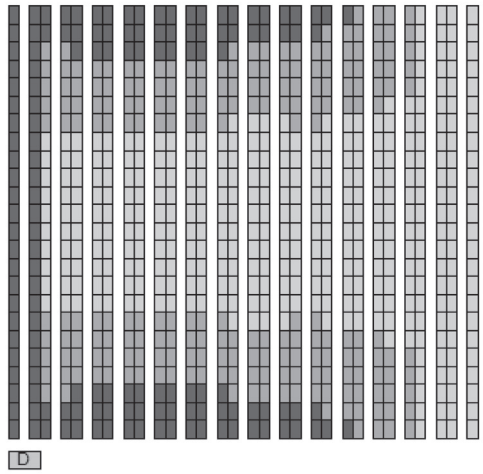
(a) Order length 2, distribution 80, 15, 5.



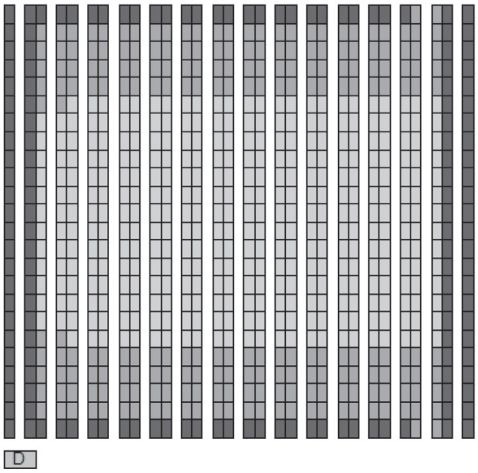
(b) Order length 2, distribution 50, 30, 20.



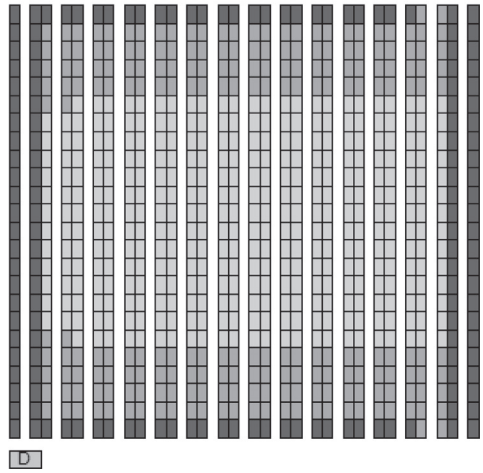
(c) Order length 10, distribution 80, 15, 5.



(d) Order length 10, distribution 50, 30, 20.



(e) Order length 20, distribution 80, 15, 5.



(f) Order length 20, distribution 50, 30, 20.

Fig. 7. Storage location assignments for midpoint routing produced by the DP for different expected order lengths and demand distributions.

From the results in Tables 1–4, it can be seen that the dynamic program consistently outperforms the common storage location assignment rules. It must be noted that solutions of the dynamic program as presented in the table for return routing are optimal. For S-shape routing, solutions of the dynamic program may not be optimal. However, due to the limitations on the solution space imposed by Corollary 2, we were able to determine optimal solutions through complete enumeration for settings with 5 aisles under S-shape routing within an hour per instance. This allowed us to benchmark at least some of these smaller instances. From a test set with 80 such instances with 5 aisles, only one instance was not solved to optimality by the dynamic program, with a gap in route length of 0.02% (Appendix D).

The results in the four tables show that in most cases one of the four common storage location assignment rules has a performance that is quite close to our dynamic program. However, which common rule is best, differs per setting. For an implementation in practice, a choice to settle for simply checking all four common rules and subsequently choosing the best, may provide acceptable performance. With our dynamic program, we are now able to determine or estimate the loss of efficiency for this option. For S-shape routing the difference of this approach with our near-optimal results is minimal. For return routing the deviation of this approach from our optimal results amounts to at most 1.82% in the 15 aisles, 2 picks, 80/15/5 pick percentage setting. Our (heuristic) solutions for largest-gap and midpoint both still improve over the best common storage location assignment rule by up to 8.4% (15 aisles, 10 picks, 50/30/20).

In Figs. 4–7 we have depicted the best storage patterns for the four routing methods under various conditions. Petersen and Schmenner (1999) note that return routing works best with diagonal storage and across-aisle storage, which is confirmed by the optimal solutions in Fig. 4. Though it must be noted that most of the optimal solutions are not diagonal or across-aisle, but rather some sort of interpolation between the two.

Jarvis and Mcdowell (1991) showed that the optimal storage location assignment is the within-aisle storage for S-shape routing. However, they approximate the length of the turn in the last aisle. Under our exact formulation from Section 6, we find slightly different results. In Fig. 5(a) and (b), we can see that the B-items in aisles 8 and 9 are not located according to within-aisle storage. Similarly, in Fig. 5(a), the A-items in aisles 2 and 3 are not stored according to within-aisle storage. In a larger set of experiments we found differences between the solution of the dynamic program and the within-aisle configuration of up to 2%. This difference comes mostly from situations where there is a high probability on orders that contain only 1 item, which pushes the preferred storage location assignment a little towards configurations that work well with return routing.

The solution shapes for largest-gap routing and midpoint routing are quite similar, as can be seen in Figs. 6 and 7. The largest differences can be observed when the demand distribution is 80/15/5; the solutions for midpoint routing appear to mix the different classes more. Petersen and Schmenner (1999) note that within-aisle and perimeter storage work best with largest gap routing. This appears consistent with the results in Fig. 6. However, this is not entirely true. In standard perimeter storage the A-items are located at the perimeter of the area. The solutions of the dynamic program show many solutions where the C-items are in the right-most aisles, and a perimeter-style distribution is used only in some of the aisles on the left. Apparently, the probability of having any *C*-item in an order in these settings is small enough to store an entire aisle of *A*-items in the middle of the warehouse and risk having to enter it twice.

Finally, the performance of the routing methods for our instances can be compared in Table 5. Midpoint routing is a restricted version of largest-gap routing. Hence, the expected route length of largest-gap routing dominates the expected route length of midpoint routing, which is confirmed by our results. Return routing is the best routing method for instances with a small expected order length, whereas largest-gap routing is better when the expected order length is larger. For random storage, Hall (1993) reports largest-gap routing to be better than S-shape routing when the number of picks per aisle is less than 3.8. The expected number of picks in each aisle in our instances is substantially lower than 3.8. Even though we evaluate storage assignments designed specifically for the routing method employed, we feel this adequately explains why S-shape routing never has the shortest expected route length in our instances.

Table 5
Expected route length for the DP solutions for the different routing methods. Minimum per row in bold.

Aisles	Picks per route	Pick percentage	Return	S-shape	Largest gap	Midpoint
7	2	80/15/5	**34.30**	45.89	45.07	45.10
7	2	50/30/20	**49.14**	57.76	54.58	54.63
7	10	80/15/5	89.56	92.82	**79.98**	80.21
7	10	50/30/20	134.89	136.28	**108.49**	109.40
7	20	80/15/5	127.34	119.69	**98.32**	98.62
7	20	50/30/20	195.37	172.01	**140.47**	143.57
15	2	80/15/5	**49.92**	60.88	58.35	58.43
15	2	50/30/20	**68.79**	77.14	72.10	72.15
15	10	80/15/5	128.40	144.33	**118.44**	118.57
15	10	50/30/20	183.89	206.11	**151.43**	152.07
15	20	80/15/5	190.29	191.44	**152.01**	152.46
15	20	50/30/20	282.91	287.45	**209.25**	211.63

10. Concluding remarks

In this paper we have presented formulas that give the exact average route length for multi-aisle multi-item picking in warehouses under the return, S-shape, largest-gap and midpoint routing methods. Our formulas work for any storage location assignment. Previous research provided formulas that were either approximations for average route length, were applicable only for a limited set of storage location patterns, or both. The exactness of the calculations for average route length is important, since we have been aiming towards proven optimal solutions. To enable fast calculations, we have derived a number of optimality conditions.

We presented a dynamic programming algorithm for determining solutions to the storage location assignment problem, which uses the optimality conditions for limiting the considered solution space and uses the exact average route length formulas as objective function. The dynamic programming algorithm is shown to provide optimal solutions under the return routing method, and near-optimal solutions under the S-shape routing method. Deviations found for solutions under S-shape routing are all less than 0.02% from optimal. The same dynamic programming algorithm is used for determining storage location assignments for largest-gap and midpoint routing. The solutions found by the dynamic program under these two routing methods, outperform common storage location assignment configurations on every considered instance. Storage location assignment patterns found by the dynamic program deviate from previously considered patterns in several ways.

Acknowledgment

This research has been funded by Dinalog, the Dutch Institute for Advanced Logistics.

Appendix A. Proof of Theorem 1

Proof. First observe that the allocation of products within an aisle does not affect the cross-aisle distance. Then Eq. (2) is equivalent to Theorem 5.1 of Eisenstein (2008). To prove Eq. (3) we will first introduce some notation. Let $F_i(j)$ be the expected travel distance in aisle i, conditioned on the fact that no location after location j is visited. Then we can describe $F_i(j)$ by the recurrence relation:

$$F_i(j) = \begin{cases} 2\left(w_c + \frac{1}{2}f\right)p_{i1} & \text{if } j = 1 \\ 2\left(w_c + \left(j - \frac{1}{2}\right)f\right)p_{ij} + (1 - p_{ij})F_i(j-1) & \text{if } 1 < j \leqslant n. \end{cases}$$

Note that $F_i(n) = \mathbb{E}(\mathbb{L}_i^*)$. Suppose there is an optimal solution A^* such that Eq. (3) does not hold. That is, there exist aisles h and i such that $h < i$ and $p_{hk}^* < p_{ik}^*$ for some $k \in \{1, \ldots, n\}$ in this optimal solution. Now construct a new solution A' by ordering the products from aisles h and i by location, that is $p'_{hj} = \max\{p_{hj}^*, p_{ij}^*\}$ and $p'_{ij} = \min\{p_{hj}^*, p_{ij}^*\}$ for all j. Let $F_i^*(j)$ and $F_i'(j)$ be the corresponding expected travel distances. We now show by mathematical induction that:

$$\Delta(j) \equiv F_h^*(j) + F_i^*(j) - F_h'(j) - F_i'(j) \geqslant 0 \quad \forall j \in \{1, \ldots, n\}. \tag{A.1}$$

Consider $j = 1$:

$$\Delta(1) = \left(w_c + \frac{1}{2}f\right)\left(p_{h1}^* + p_{i1}^* - \min(p_{h1}^*, p_{i1}^*) - \max(p_{h1}^*, p_{i1}^*)\right) = 0$$

Hence Eq. (A.1) hold for $j = 1$. Now, as induction hypothesis, assume Eq. (A.1) holds for $j = \hat{j}$, we then have for $j = \hat{j} + 1$

$$\Delta(\hat{j}+1) = (1 - p_{h\hat{j}+1}^*)F_h^*(\hat{j}) + (1 - p_{i\hat{j}+1}^*)F_i^*(\hat{j}) - (1 - p'_{h\hat{j}+1})F_h'(\hat{j}) - (1 - p'_{i\hat{j}+1})F_i'(\hat{j}).$$

Consider two cases.

Case 1: $p'_{h\hat{j}+1} = p_{h\hat{j}+1}^*$ and $p'_{i\hat{j}+1} = p_{i\hat{j}+1}^*$. Then some rearrangement of terms yields

$$\Delta(\hat{j}+1) = (1 - p_{h\hat{j}+1}^*)(F_h^*(\hat{j}) + F_i^*(\hat{j}) - F_h'(\hat{j}) - F_i'(\hat{j})) + (p_{h\hat{j}+1}^* - p_{i\hat{j}+1}^*)(F_i^*(\hat{j}) - F_i'(\hat{j})) \geqslant 0.$$

The inequality follows from the induction hypothesis, from $p_{h\hat{j}+1}^* \geqslant p_{i\hat{j}+1}^*$ and from $F_i^*(\hat{j}) \geqslant F_i'(\hat{j})$, which in turn follows directly from $p_{ij}^* \geqslant p'_{ij}\ \forall j \in \{1, \ldots, n\}$.

Case 2: $p'_{h\hat{j}+1} = p_{i\hat{j}+1}^*$ and $p'_{i\hat{j}+1} = p_{h\hat{j}+1}^*$. Then similar to case 1 we have

$$\Delta(\hat{j}+1) = (1 - p_{i\hat{j}+1}^*)(F_h^*(\hat{j}) + F_i^*(\hat{j}) - F_h'(\hat{j}) - F_i'(\hat{j})) + (p_{i\hat{j}+1}^* - p_{h\hat{j}+1}^*)(F_h^*(\hat{j}) - F_i'(\hat{j})) \geqslant 0,$$

since $p_{hj}^* \geqslant p'_{ij}\ \forall j \in \{1, \ldots, n\}$ implies $F_h^*(\hat{j}) \geqslant F_i'(\hat{j})$. Hence, we can conclude that Eq. (A.1) is true. This implies that the expected in-aisle travel distance for A' is less than or equal to that of A^*. Additionally A' will have a lower expected

cross-aisle travel than A^*, as is shown in the proof of Theorem 2 in Jarvis and Mcdowell (1991). This contradicts that A^* is optimal and completes the proof. □

Appendix B. Determining expected largest gap

The expected largest gap can be determined using a Dynamic Program. Before going into the specifics of the DP, we first focus on the task of determining the largest gap in aisle i for a given (deterministic) order by means of a step-wise procedure. The DP will then later calculate the expected largest gap, given that the step-wise procedure is in a certain state. The expected largest gap in the aisle is then obtained by calculating the expected largest gap given the start state of the step-wise procedure.

The idea of the procedure is to go through the aisle, keeping track of both the current gap and the largest gap found so far, updating the latter only if it is exceeded by the current gap. A key aspect is that we condition on whether there is a pick at a location or not, this enables us to expand the procedure to the stochastic case. We start our calculation in the front cross aisle and define the length of the current gap, $g_c(0)$, to be $g_c(0) = \frac{1}{2}f + w_c$, and the length of the longest gap found so far, $g_\ell(0)$, to be $g_\ell(0) = \frac{1}{2}f + w_c$. That is, being in the front cross aisle, we know we can at least move to the first location in the aisle before encountering a pick item, which is a distance of $\frac{1}{2}f + w_c$. Next we observe whether a pick item is to be made at location 1. If a pick item is at location 1, then $g_c(1) = f$ and $g_\ell(1) = \max\{g_\ell(0); f\}$. That is, the current gap $g_c(1)$ ranges from location 1 until at least location 2, and the largest observed gap $g_\ell(1)$ is either between the front cross aisle and location 1, or between locations 1 and 2. If there is no pick item at location 1, then we calculate $g_c(1) = g_c(0) + f$ and $g_\ell(1) = g_\ell(0) + f$. That is, we know the current gap ranges from the front cross aisle at least until location 2. Generally, moving from location $j-1$ to location j, for $j = 1, \ldots, n-1$ we update

$$g_c(j) = \begin{cases} f & \text{if there is a pick item at location } j \\ g_c(j-1) + f & \text{if there is no pick item at location } j, \end{cases}$$

and

$$g_\ell(j) = \begin{cases} \max\{g_\ell(j-1); f\} & \text{if there is a pick item at location } j \\ \max\{g_\ell(j-1); g_c(j-1) + f\} & \text{if there is no pick item at location } j. \end{cases}$$

For the last location n we have:

$$g_c(n) = \begin{cases} \frac{1}{2}f + w_c & \text{if there is a pick item at location } n \\ g_c(n-1) + \frac{1}{2}f + w_c & \text{if there is no pick item at location } n, \end{cases}$$

and

$$g_\ell(n) = \begin{cases} \max\left\{g_\ell(n-1); \frac{1}{2}f + w_c\right\} & \text{if there is a pick item at location } n \\ \max\left\{g_\ell(n-1); g_c(n-1) + \frac{1}{2}f + w_c\right\} & \text{if there is no pick item at location } n. \end{cases}$$

The largest gap in this aisle for a deterministically given set of pick items is then given by $g_\ell(n)$. Next, we add probabilities and perform backward stochastic dynamic programming on the described structure to obtain the *expected* largest gap. We define states (g_c, g_l) for all possible combinations of gap lengths. Note that since states are invariant, we omit the indexing (j), we used above. Even though we start with a large number of states, this amount decreases in each step, which ensures the process to be fast to calculate. Let the expected largest gap in aisle i, given that the state at location j of the step-wise procedure formulated above is (g_c, g_ℓ), be denoted $G_{ij}(g_c, g_\ell)$.

Example 1. Consider the aisle in Fig. B.8. The aisle has 8 picks, the distances are $w_c = 3$ and $f = 1$. One can see directly from the picture that the gaps in the aisle are 4.5 at the beginning, 3 in the middle and 6.5 at the end, leading to a largest gap of 6.5. As an illustration, the values of the step-wise procedure are given in Table D.6. Below, the step-wise procedure is executed step by step for all locations.

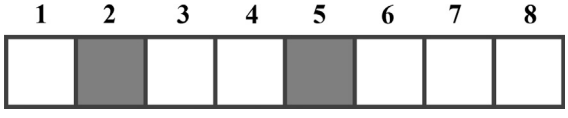


Fig. B.8. Aisle Example 1.

Table D.6
Current and largest gap for aisle Example 1.

Step	0	1	2	3	4	5	6	7	8
g_c	3.5	4.5	1	2	3	1	2	3	6.5
g_ℓ	3.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	6.5

0. Starting before the first pick at the front of the aisle we have a gap of $w_c + 0.5f = 3.5$, hence $g_c(0) = 3.5$ and $g_\ell(0) = 3.5$.
1. At location 1 we have no pick, increasing the current gap by 1 to $g_c(1) = 4.5$. Moreover, for the largest gap we have $g_\ell(1) = \max\{g_\ell(0), g_c(0) + 1\} = 4.5$.
2. There is a pick at location 2, resetting the current gap to $g_c(2) = 1$. Furthermore, $g_\ell(2) = \max\{g_\ell(1), 1\} = 4.5$.
3. No pick at location 3, hence $g_c(3) = g_c(2) + 1 = 2$ and $g_\ell(3) = \max\{g_\ell(2), g_c(2) + 1\} = 4.5$.
4. No pick at location 4, hence $g_c(4) = g_c(3) + 1 = 3$ and $g_\ell(4) = \max\{g_\ell(3), g_c(3) + 1\} = 4.5$.
5. A pick at location 5, hence $g_c(5) = 1$, and $g_\ell(5) = \max\{g_\ell(4), 1\} = 4.5$.
6. No pick at location 6, hence $g_c(6) = g_c(5) + 1 = 2$ and $g_\ell(6) = \max\{g_\ell(5), g_c(5) + 1\} = 4.5$.
7. No pick at location 7, hence $g_c(7) = g_c(6) + 1 = 3$ and $g_\ell(7) = \max\{g_\ell(6), g_c(6) + 1\} = 4.5$.
8. No pick at location $8 = n$, hence $g_c(8) = g_c(7) + \frac{1}{2}f + w_c = 6.5$. Moreover, $g_\ell(8) = \max\{g_\ell(7), g_c(7) + 3.5\} = 6.5$.

Finally, the largest gap in the aisle is obtained as $g_\ell(8) = 6.5$.

Step 1: As is common with dynamic programming we start in a trivial stage, i.e., beyond the last location in location "n + 1". Available states (g_c, g_ℓ) are such that $g_c, g_\ell \in \{\frac{1}{2}f + w_c, 1\frac{1}{2}f + w_c, \ldots, (n - \frac{1}{2})f + w_c, nf + 2w_c\} \cup \{f, 2f, 3f, \ldots, (n-1)f\}$, with $g_c \leqslant g_\ell$. The expected largest gap $G_{i,n+1}$ for each state is set by definition equal to the state's value for the largest gap g_ℓ, i.e., $G_{i,n+1}(g_c, g_\ell) \equiv g_\ell$.

Step 2: Now we consider location n. We observe the probability p_{in} for location n and determine $G_{in}(g_c, g_\ell)$ for all possible states (g_c, g_ℓ) with $g_c, g_\ell \in \{\frac{1}{2}f + w_c, 1\frac{1}{2}f + w_c, \ldots, (n - \frac{1}{2})f + w_c\} \cup \{f, 2f, 3f, \ldots, (n-1)f\}$ and $g_c \leqslant g_\ell$ as:

$$G_{in}(g_c, g_\ell) = (1 - p_{in})G_{i,n+1}\left(g_c + \frac{1}{2}f + w_c, \max\left\{g_\ell; g_c + \frac{1}{2}f + w_c\right\}\right) + p_{in}G_{i,n+1}\left(\frac{1}{2}f + w_c, \max\left\{g_\ell; \frac{1}{2}f + w_c\right\}\right).$$

Step 3: For locations $j = 1, \ldots, n-1$, we determine $G_{ij}(g_c, g_\ell)$ for all possible states (g_c, g_ℓ) with $g_c, g_\ell \in \{\frac{1}{2}f + w_c, 1\frac{1}{2}f + w_c, \ldots, (j - \frac{1}{2})f + w_c\} \cup \{f, 2f, 3f, \ldots, (j-1)f\}$ and $g_c \leqslant g_\ell$ as:

$$G_{ij}(g_c, g_\ell) = (1 - p_{ij})G_{i,j+1}(g_c + f, \max\{g_\ell; g_c + f\}) + p_{ij}G_{i,j+1}(f, \max\{g_\ell; f\}). \tag{B.1}$$

Step 4: Finally, to obtain the expected largest gap G_i for aisle i note that

$$G_i \equiv G_{i0}\left(\frac{1}{2}f + w_c, \frac{1}{2}f + w_c\right)$$

Appendix C. Storage assignments

In this section we explain how to generalize pre-defined storage assignments for class-based storage to full-turnover storage. Not only does this offer guidance when storing individual products based on full turnover, but it also helps when the storage classes do not 'fit' in the shape some storage assignments assume, e.g., the number of products in a class is not a multiple of the number of products in an aisle. First, the products are ordered based on their picking probability. In the case of class-based storage all products in a class have the same picking probability and are sorted randomly within their class. Secondly, the locations in the warehouse are ordered on preference. This preference depends on the storage assignment method used. Below we will specify preference functions u^w for within-aisle (WA), u^a for across-aisle (AA), u^d for diagonal (DI), and u^p for perimeter (PE) storage. Finally, products are assigned to locations according to the two orders.

Within-aisle storage confines storage classes to entire aisles. Aisles closer to the depot are more desirable and contain the classes with higher demands. Thus, the preference of a storage location (i, j) is based primarily on the aisle index i. The secondary condition determining the preference is the location index j. Hence, when locations share the same aisle, the locations closest to the front cross-aisle are preferred. Letting $\epsilon > 0$ be some small number, the following preference function reflects these properties:

$$u^w(i, j) = -i - \epsilon j. \tag{C.1}$$

Across-aisle storage is the opposite of with-in aisle storage. Now the primary condition on which the preference of a storage location is based is the location index j, whereas the aisle i is the secondary condition. Hence we obtain similarly

$$u^a(i,j) = -\epsilon i - j. \tag{C.2}$$

Diagonal storage is based on the distance to the depot. Locations closer to the depot are preferred over locations further away. Recall that the distance from the depot for a location (i,j) is given by $w_a(i-1) + w_c + f(j-\frac{1}{2})$. Simplifying the distance function by removing constants, we obtain the following preference function

$$u^d(i,j) = -w_a i - fj. \tag{C.3}$$

Finally, for perimeter storage locations closer to the perimeter are preferred over locations further away. Therefore, aisles 1 and m are preferred over all other aisles, where aisle 1 is preferred over aisle m. Due to Corollary 3 we order the locations in these aisle by their distance from the front cross-aisle. For other aisles, the distance to the cross-aisles is the primary condition on which locations are ordered. The aisle number is the secondary condition. Finally, the distance to the front cross-aisle is used as a third condition when both the primary and secondary condition are the same for two locations. Let M be some large number. Then the following preference function represents the preferences described above.

$$u^p(i,j) = \begin{cases} -\epsilon j - i & \text{if } i = 1,\ i = m \\ -M - \min\{\epsilon^2 + n + 1 - j, j\} - \epsilon i & \text{if } 1 < i < m. \end{cases} \tag{C.4}$$

Appendix D. Optimal solutions S-shape routing

See Table D.7.

Table D.7
Objective value of the optimal solution and our DP for instances with 5 aisles and S-shape routing.

n	x_A	x_B	x_C	p_A	p_B	p_C	Our DP	Optimal	Difference (%)
16	16	24	40	0.05000	0.00625	0.00125	20.801	20.797	0.019
16	16	24	40	0.03125	0.01250	0.00500	27.876	27.876	0.000
16	16	24	40	0.10000	0.01250	0.00250	34.642	34.642	0.000
16	16	24	40	0.06250	0.02500	0.01000	49.115	49.115	0.000
16	16	24	40	0.25000	0.03125	0.00625	56.586	56.586	0.000
16	16	24	40	0.15625	0.06250	0.02500	87.131	87.131	0.000
16	16	24	40	0.50000	0.06250	0.01250	75.506	75.506	0.000
16	16	24	40	0.31250	0.12500	0.05000	116.978	116.978	0.000
16	16	24	40	1.00000	0.12500	0.02500	99.033	99.033	0.000
16	16	24	40	0.62500	0.25000	0.10000	141.333	141.333	0.000
24	24	36	60	0.03333	0.00417	0.00083	26.597	26.597	0.000
24	24	36	60	0.02083	0.00833	0.00333	34.434	34.434	0.000
24	24	36	60	0.06667	0.00833	0.00167	44.985	44.985	0.000
24	24	36	60	0.04167	0.01667	0.00667	61.321	61.321	0.000
24	24	36	60	0.16667	0.02083	0.00417	72.493	72.493	0.000
24	24	36	60	0.10417	0.04167	0.01667	108.949	108.949	0.000
24	24	36	60	0.33333	0.04167	0.00833	95.436	95.436	0.000
24	24	36	60	0.20833	0.08333	0.03333	146.894	146.894	0.000
24	24	36	60	0.66667	0.08333	0.01667	124.513	124.513	0.000
24	24	36	60	0.41667	0.16667	0.06667	178.580	178.580	0.000
16	24	24	32	0.03333	0.00625	0.00156	24.025	24.025	0.000
16	24	24	32	0.02083	0.01250	0.00625	30.658	30.658	0.000
16	24	24	32	0.06667	0.01250	0.00313	41.881	41.881	0.000
16	24	24	32	0.04167	0.02500	0.01250	53.894	53.894	0.000
16	24	24	32	0.16667	0.03125	0.00781	71.464	71.464	0.000
16	24	24	32	0.10417	0.06250	0.03125	96.183	96.183	0.000
16	24	24	32	0.33333	0.06250	0.01563	92.589	92.589	0.000
16	24	24	32	0.20833	0.12500	0.06250	126.752	126.752	0.000
16	24	24	32	0.66667	0.12500	0.03125	112.812	112.812	0.000
16	24	24	32	0.41667	0.25000	0.12500	147.433	147.433	0.000
24	36	36	48	0.02222	0.00417	0.00104	30.291	30.291	0.000
24	36	36	48	0.01389	0.00833	0.00417	37.563	37.563	0.000
24	36	36	48	0.04444	0.00833	0.00208	53.034	53.034	0.000
24	36	36	48	0.02778	0.01667	0.00833	66.639	66.639	0.000
24	36	36	48	0.11111	0.02083	0.00521	90.206	90.206	0.000
24	36	36	48	0.06944	0.04167	0.02083	120.202	120.202	0.000
24	36	36	48	0.22222	0.04167	0.01042	117.260	117.260	0.000
24	36	36	48	0.13889	0.08333	0.04167	159.532	159.532	0.000
24	36	36	48	0.44444	0.08333	0.02083	142.829	142.829	0.000
24	36	36	48	0.27778	0.16667	0.08333	186.833	186.833	0.000

References

Caron, F., Marchet, G., Perego, A., 1998. Routing policies and COI-based storage policies in picker-to-part systems. Int. J. Prod. Res. 36 (3), 713–732.
Chen, C.M., Gong, Y., De Koster, R.B.M., van Nunen, J.A.E.E., 2010. A flexible evaluative framework for order picking systems. Prod. Oper. Manage. 19 (1), 70–82.
Chew, E.P., Tang, L.C., 1999. Travel time analysis for general item location assignment in a rectangular warehouse. Eur. J. Oper. Res. 112 (3), 582–597.
Chou, Y.C., Chen, Y.H., Chen, H.M., 2012. Recency-based storage assignment and warehouse configuration for recurrent demands. Comput. Ind. Eng. 62 (4), 880–889.
De Koster, R.B.M., Le-Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: a literature review. Eur. J. Oper. Res. 182 (2), 481–501.
Dekker, R., de Koster, M.B.M., Roodbergen, K.J., van Kalleveen, H., 2004. Improving order-picking response time at Ankor's warehouse. Interfaces 34 (4), 303–313.
Eisenstein, D.D., 2008. Analysis and optimal design of discrete order picking technologies along a line. Naval Res. Logist. 55 (4), 350–362.
Ene, S., Öztürk, N., 2011. Storage location assignment and order picking optimization in the automotive industry. Int. J. Adv. Manuf. Technol. 60 (5–8), 787–797.
Frazelle, E.H., Hackman, S.T., Passy, U., Platzman, L.K., 1994. The forward-reserve problem. In: Ciriani, Tito A., Leachman, Robert C. (Eds.), Optimization in industry 2. John Wiley & Sons, pp. 43–61.
Gu, J., Goetschalckx, M., McGinnis, L.F., 2007. Research on warehouse operation: a comprehensive review. Eur. J. Oper. Res. 177 (1), 1–21.
Hall, R.W., 1993. Distance approximations for routing manual pickers in a warehouse. IIE Trans. 25 (4), 76–87.
Hausman, W.H., Schwarz, L.B., Graves, S.C., 1976. Optimal storage assignment in automatic warehousing systems. Manage. Sci. 22 (6), 629–638.
Heragu, S.S., Du, L., Mantel, R.J., Schuur, P.C., 2005. Mathematical model for warehouse design and product allocation. Int. J. Prod. Res. 43 (2), 327–338.
Hsieh, L.-f., Tsai, L., 2005. The optimum design of a warehouse system on order picking efficiency. Int. J. Adv. Manuf. Technol. 28 (5–6), 626–637.
Hwang, H., Oh, Y.H., Lee, Y.K., 2004. An evaluation of routing policies for order-picking operations in low-level picker-to-part system. Int. J. Prod. Res. 42 (18), 3873–3889.
Hwang, H.S., Cho, G.S., 2006. A performance evaluation model for order picking warehouse design. Comput. Ind. Eng. 51 (2), 335–342.
Jarvis, J.M., Mcdowell, E.D., 1991. Optimal product layout in an order picking warehouse. IIE Trans. 23 (1), 93–102.
Jewkes, E., Lee, C., Vickson, R., 2004. Product location, allocation and server home base location for an order picking line with multiple servers. Comput. Oper. Res. 31 (4), 623–636.
Kunder, R., Gudehus, T., 1975. Mittlere wegzeiten beim eindimensionalen kommissionieren. Z. Oper. Res. 19, 53–72.
Le-Duc, T., De Koster, R.B.M., 2005. Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse. Int. J. Prod. Res. 43 (17), 3561–3581.
Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. 23, 1–2.
Muppani, V.R., Adil, G.K., 2008. A branch and bound algorithm for class based storage location assignment. Eur. J. Oper. Res. 189 (2), 492–507.
Pan, J.C.H., Shih, P.H., Wu, M.H., 2012. Storage assignment problem with travel distance and blocking considerations for a picker-to-part order picking system. Comput. Ind. Eng. 62 (2), 527–535.
Pan, J.C.-H., Shih, P.-H., Wu, M.-H., Lin, J.-H., 2015. A storage assignment heuristic method based on genetic algorithm for a pick-and-pass warehousing system. Comput. Ind. Eng. 81, 1–13.
Parikh, P.J., Meller, R.D., 2010. A travel-time model for a person-onboard order picking system. Eur. J. Oper. Res. 200 (2), 385–394.
Petersen, C.G., Schmenner, R.W., 1999. An evaluation of routing and volume-based storage policies in an order picking operation. Decis. Sci. 30 (2), 481–501.
Rao, S.S., Adil, G.K., 2013a. Class-based storage with exact S-shaped traversal routeing in low-level picker-to-part systems. Int. J. Prod. Res. 51 (16), 4979–4996.
Rao, S.S., Adil, G.K., 2013b. Optimal class boundaries, number of aisles, and pick list size for low-level order picking systems. IIE Trans. 45 (12), 1309–1321.
Ratliff, H.D., Rosenthal, A.S., 1983. Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. Oper. Res. 31 (3), 507–521.
Roodbergen, K.J., Vis, I.F.A., 2006. A model for warehouse layout. IIE Trans. 38 (10), 799–811.
Theys, C., Bräysy, O., Dullaert, W., Raa, B., 2010. Using a TSP heuristic for routing order pickers in warehouses. Eur. J. Oper. Res. 200 (3), 755–763.
Thomas, L.M., Meller, R.D., 2014. Analytical models for warehouse configuration. IIE Trans. 46 (9), 928–947.
Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A., 2003. Facilities Planning. John Wiley & Sons.