Otto von Guericke University
Magdeburg

Order Picking: A Survey of Planning Problems and Methods

Gerhard Wäscher

Working Paper No. 13/2002

FEMM

Faculty of Economics and Management
Magdeburg

Working Paper Series

Order Picking: A Survey of Planning Problems and Methods

Gerhard Wäscher

Working Paper No. 13/2002

Impressum:

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

V. i. S. d. P.: Gerhard Wäscher

Fakultät für Wirtschaftswissenschaft
Lehrstuhl für Betriebswirtschaftslehre
-Management Science-
Otto-von-Guericke-Universität Magdeburg
Postfach 4120
39016 Magdeburg

Auflage: 76
Redaktionsschluß: November 2002

Herstellung: Dezernat Allgemeine Angelegenheiten,
Sachgebiet Reproduktion

Bezug über den Herausgeber

ISSN 1615-4274

Order Picking:
A Survey of Planning Problems and Methods

Gerhard Wäscher

Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Lehrstuhl für Betriebswirtschaftslehre
- Management Science -
Postfach 4120
39016 Magdeburg

Abstract: Order picking – as a warehouse function – is often considered to be critical for the public manifestation of a supply chain. Underperformance can result both in unsatisfactory customer service and in high cost. In this paper the author will give an overview of planning problems and corresponding methods which have been suggested in the literature for the reduction of cost and the improvement of customer service in picker-to-product systems. In particular, the following central issues will be addressed: Item location, order batching, and picker routing.

Keywords: Key Words: Order Picking, Warehouse Management, Location, Batching, Routing

Table of Contents

1. Introduction

Order picking is a warehouse function critical to each supply chain. Underperformance results in unsatisfactory customer service (long processing and delivery times, incorrect shipments) and high costs (labour cost, cost of additional and/or emergency shipments), both being a significant threat to the competitiveness of the total chain. Despite its criticality, order picking is not really a topic of major interest, neither in academic research nor in industrial practice. Like other warehousing operations, it appears to be one of the most frequently overlooked, underfunded, and inadequately planned corporate functions (Tompkins et al. 1996). Furthermore, managers seem to be unaware of advanced planning techniques and their potential with respect to reducing cost and improving customer service (Petersen 1999, p. 1054).

The aim of this paper is to review these planning techniques and to demonstrate what benefits can be expected from their use in practice. Due to limitations of space, the focus of the paper will be on conventional (manual) order-picking systems, for which the main operational planning issues will be introduced, namely the assignment of items to storage locations, the transformation of customer orders into picking orders, and the routing of pickers through the warehouse. Corresponding planning techniques will be presented and discussed.

2. Order Picking as a Warehouse Function

Order picking can be defined as the retrieval of items from their warehouse locations in order to satisfy demands from (internal or external) customers (Petersen/Schmenner 1999, p. 481). As a warehouse function order picking arises because incoming articles are received and stored in (large-volume) unit loads while customers order small volumes (less-than-unit loads) of different articles. Typically, thousands of customer orders have to be processed in a distribution warehouse per day.

Even though there have been various attempts to automate the picking process, picker-less systems are rarely found in practice. Order picking – like many other material-handling activities – still is a repetitive, labour-intensive activity these days (Petersen 1999, p. 1055). Order-picking systems which involve human operators can be organized in two ways, namely as a *product-to-picker system*, in which the requested products are delivered automatically to a person at an input/output (I/O) point, or as a *picker-to-product system*, in which a person (*order picker*) travels to storage locations in order to collect the required products. Product-to-picker systems have been studied extensively (Cormier/Gunn 1992), therefore the focus of this paper will be on picking systems of the second type.

Typically, when order picking is organized as a picker-to-product system, the storage area of the warehouse is divided into two parts, namely into a *reserve (storage) area*, in which full unit loads are stored (usually on pallets), and into a *forward (picking) area,* which is specifically designed to support and facilitate the picking process of small, fast moving items and is replenished from the reserve storage area (Ballou 1967, p. 60). The picking area consists of a number of *aisles* with the different products (*items*) stored at both sides of the aisles. The operations of the order pickers may be restricted to a single aisle each (as in *person-aboard automatic storage/retrieval systems*), or they may be free to access any aisle (as in more conventional picking systems). Like it is common practice in the literature, the order-picking area of a warehouse will be addressed as the "*order-picking warehouse*" below.

3. Order-Picking System

3.1. Basic System Design and Layout

In the order-picking warehouse to be considered here, the items are stored on pallets in pallet racks. Picking is carried out from the lowest level, only. The *storage locations* (*bays*) are of identical size, each can accommodate a single pallet. They are located in a number of parallel, straight (*picking*) *aisles*, which run perpendicular to the front end of the picking area. The (picking) aisles are of equal length and width (*rectangular layout*). Changing from one (picking) aisle to another is possible by means of two *cross aisles*, one at the front and one at the rear of the warehouse. Fig. 1 depicts the layout of the picking area, which is consistent with the warehouse layout literature (e.g. Bassan/Roll/Rosenblatt 1980; Caron/Marchet/Perego 2000). More specifically, it can be characterized as a *one-block layout*. (By introduction of one or several additional cross aisles, a *multi-block layout* would be obtained.) In the literature, the layout of Fig. 1 is often considered as the *basic layout* of an order-picking area (Roodbergen/de Koster 2001b, p. 32).

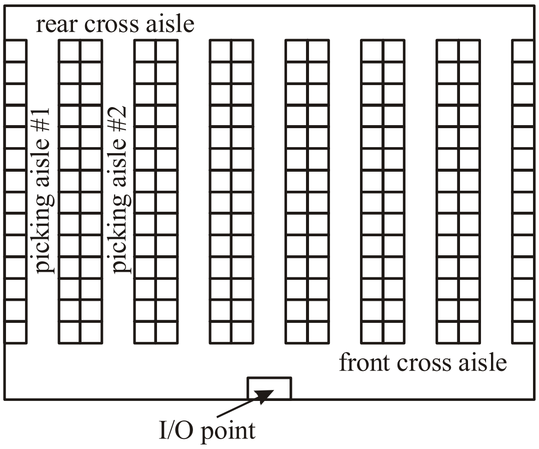


Fig. 1: Picking-area layout

The *input/output* (*I/O*) *point* (also called *depot* or *dock*) defines a location where the order picker enters the picking area, and where she/he afterwards returns to in order to deposit the picked items. It will be assumed that the I/O point is located in the middle of the front end of the warehouse.

The basis of the actual picking process is provided by a set of *customer orders*, each one consisting of a number of order lines. An *order line* or *position* represents a product type or item type (identified by an article number) and the corresponding quantity, which has been requested by the customer. The set of customer orders is assumed to be fixed and known in advance (*off-line problem*). Usually, the customer orders will not be processed one by one. Instead, at first, the set of customer orders is rearranged into a set of *picking orders*. A picking order may consist of a subset of the positions from a customer order (in particular, if the customer order is large), it may comprise several complete customer orders (in case of relatively small customer orders), or it may contain subsets of positions from several customer orders.

The actual picking process starts when the order picker receives a *pick list* at the I/O point. Each pick list is a picking order to which information has been added about

- the locations where the items are stored, and
- the sequence according to which the locations are to be visited.

In order to pick the requested items, the order picker collects a device such as a small picking truck or a roll cage. Then she/he walks or rides to the respective storage locations (*pick locations*) and retrieves the products in the necessary quantities. The items are packed on the picking device, which facilitates transportation of the picked items, but also enables her/him to collect items at several storage locations before she/he returns to the I/O point.

It is assumed that the aisles are wide enough to allow for overtaking and two-way-travel. Items can be picked from both sides of an aisle in a single move (i.e. the picker does not have to change her/his position significantly) such that the horizontal distance between the racks in an aisle can be neglected (*"narrow" aisles*). These characteristics define a *standard order-picking warehouse* to which we will refer to unless otherwise stated.

3.2. Planning Issues Concerning Policies and Operations

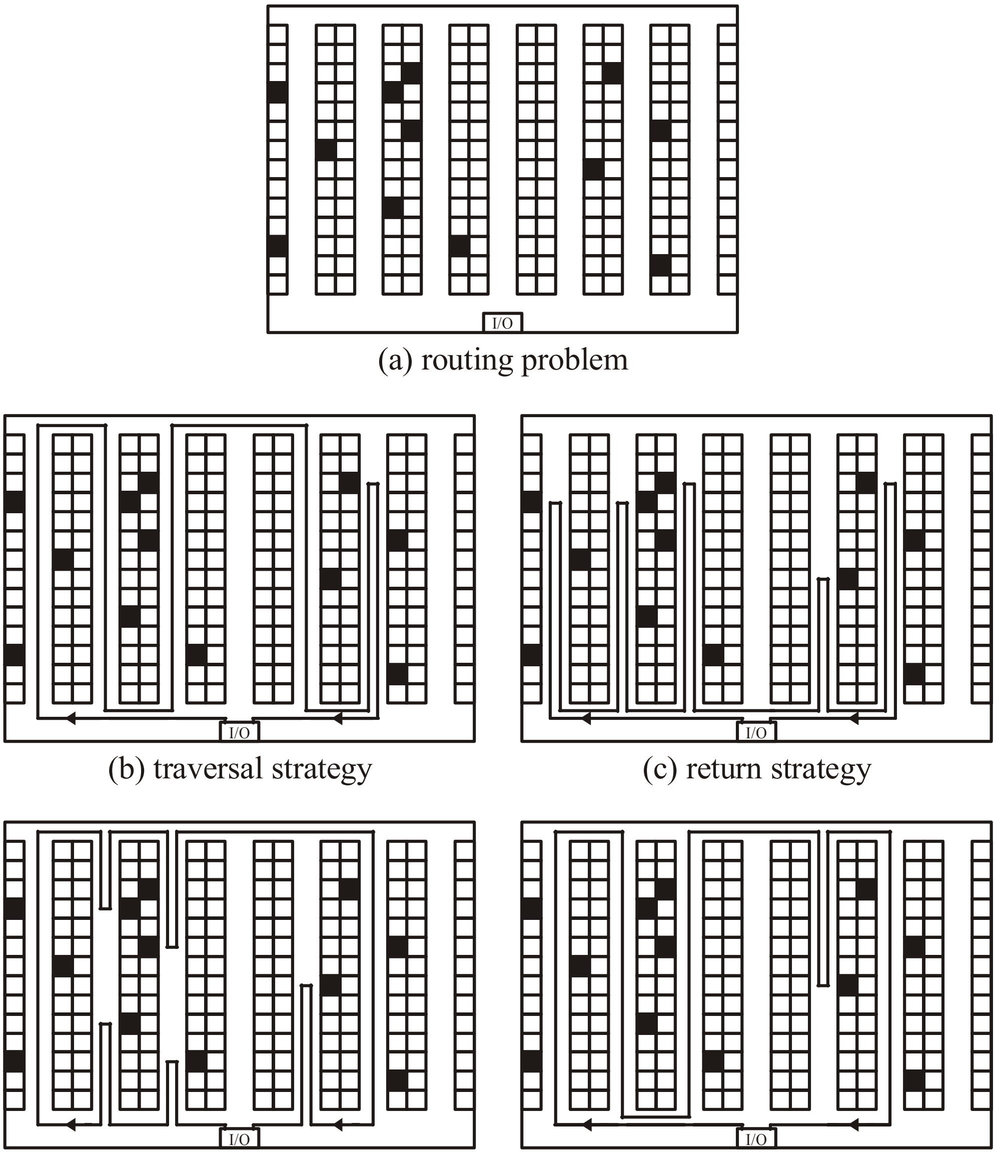
For this review, a given order-picking warehouse of the standard type will be assumed. Design issues, such as questions of layout optimization, will not be discussed, here. Instead, we refer to the literature (e.g. Caron/Marchet/Perego 2000, Roodbergen 2001, Vaughan/Petersen 1999). The remaining planning issues have to be addressed on two levels, on the level of policies and on the level of operations. They are related to three main complexes (Caron/Marchet/Perego 1998),

- the assignment of items to storage locations (*storage location*),
- the transformation of customer orders into picking orders (*order consolidation*), and
- the routing of pickers through the warehouse (*picker routing*).

In general, *policies* can be looked upon as basic principles according to which processes are organized. They define the framework for subsequent operational decisions. A *storage policy* determines how storage locations are allocated to products. In *dedicated storage* all products are stored at fixed locations. Each article can be found in the same location for a relatively long period of time. *Randomized storage* is a storage policy in which an incoming pallet is assigned to one of the currently available, empty locations. Over time, the same article will be found in different locations. Obviously, this kind of storage policy requires a computer-controlled environment, which supports the order picker in identifying the locations of the requested articles. Dedicated and randomized storage are combined in *class-based storage*. According to this policy, the articles are divided into several classes at first, each of which is then assigned to a dedicated area of the warehouse.

Within the framework of dedicated storage it has to be decided where the various item types should be stored (*item-location problem*) while in connection with a randomized storage policy the decision problem arises to which of the available locations an incoming lot of an article should be assigned. Class-based storage requires deciding into how many classes the articles should be arranged and where the classes should be located in the warehouse.

The *order consolidation policy* is the principle according to which customer orders are rearranged into picking orders. In *single order picking* (also called *strict order picking*; Ruben/Jacobs 1999, p. 576) each customer order is directly taken as a picking order. Alternatively, when the order size is small in relation to the capacity of the transportation device, *order batching* may be applied. In this case, several customer orders are combined into a *batch*, i.e. a single picking order, which is retrieved by a single picker. The option of batching customer orders gives rise to the question of how a given set of customer orders should be combined into smaller subsets each of which representing a picking order (*order-batching problem*).



(a) routing problem

(b) traversal strategy

(c) return strategy

(d) largest-gap strategy

(e) composite strategy

Fig. 2: Routing strategies for a one-block layout

A *routing policy* is a principle for the design of picking tours through the warehouse. *Individual routing* means that for each picking order it has to be decided in which sequence the various pick locations are to be visited (*picker-routing problem*). In other words, within the framework of individual routing an individual, usually optimal tour has to be developed for each picking order. In practice, frequently *standardized routing* is applied. The picking tour that corresponds to a particular picking order is then developed on the basis of a simple *rout-*

ing strategy. (A brief description of such routing strategies can be found, e.g., in Petersen/Schmenner 1999, pp. 482-487.) The most commonly-known strategies are the so-called traversal, return, largest-gap and composite strategies (see Fig. 2). They have in common that only *pick-location aisles* are visited, i.e. aisles, which contain at least one pick location. The strategies will be introduced for one-block layouts only, here. As for modifications for multi-block layouts, we refer to the literature (e.g. Roodbergen 2001, pp. 38 ff.)

The black-marked bays of Fig. 2a indicate a set of *pick locations*, i.e. locations of items, which have to be picked according to a given picking order (for this particular picker-routing problem see de Koster/van der Poort 1998, p. 471). Application of the *traversal strategy* (see Fig. 2b), also called *S-shape strategy* (e.g. see Roodbergen 2001, p. 33), generates a tour in which each pick-location aisle is traversed completely (with the exception of the last one, if an odd number of pick-location aisles has to be visited). The picker starts at the I/O point, walks/rides to the left-most pick-location aisle, enters it from the front and leaves it at the rear, then proceeds to the next pick-location aisle, which is traversed from the rear to the front, etc., until the last item has been picked from which location the picker returns to the I/O point.

When the *return strategy* (see Fig. 2c) is applied, the order picker enters and leaves all pick-location aisles from the same cross aisle (i.e. usually from the front cross aisle). She/he traverses each pick-location aisle up the farthest pick location and returns from there to the cross aisle.

According to the *largest-gap strategy* (see Fig. 2d), return trips into pick-location aisles can be performed from both the front and the rear cross aisle. At which location the picker has to turn back in a pick-location aisle is determined by the largest gap. A gap represents a separation between the locations of two adjacent items to be picked (*picks*) in a picking aisle, or between the location of the first pick in a picking aisle and the front cross aisle, or between the location of the last pick in a picking aisle and the rear cross aisle. The picker takes a return trip from both the front and the rear cross aisle if the largest gap is between two adjacent picks. Otherwise, a return trip from either the front or the rear cross aisle is used. In each case, the largest gap of a pick-location aisle represents the portion of the aisle that will not be traversed by the order picker (Petersen/Schmenner, 1999, pp. 485 f.). The left-most aisle that has to be visited will be traversed completely in order to reach the rear cross aisle. Likewise, in order to make her/his way back to the depot, the order picker will pass through the right-most pick-location aisle completely from the rear to the front.

The *composite strategy* (Petersen 1997, p. 1102; see Fig. 2e) combines elements from both traversal and return strategy. It seeks to minimize the distance between the farthest pick locations in two adjacent aisles (Petersen/Schmenner 1999, pp. 486 f.).

3.3. Planning Goals

Within the framework of a given (standard) order-picking warehouse, minimization of the total time necessary to process a given set of customer orders, or, equivalently, to pick all items of a given set of customer orders (*total processing time, total picking time*) is the central goal for planning and controlling the picking processes. On one hand, a reduction of the total picking time corresponds to an increase in the speed of the picking activities, resulting in shorter delivery times. This, in turn, will improve the customer service. Likewise, on the other hand, a reduction in the total processing time leads to a smaller demand of man-power (order pickers). In the short run, labour costs related to overtime, temporary staff etc. can be reduced. In the long run, the necessary number of permanent staff may also decrease.

Improving the performance and reducing the cost of the system therefore can be achieved by reducing the total picking time (Caron/Marchet/Perego 2000, p. 101). From this point of view it is worthwhile to analyse what components make up for the total picking time and how these components can be influenced by the above-described decisions. The time necessary to complete picking of a customer/picking order includes (Petersen 1999, p. 1055)

- the time needed at the I/O point for performing the necessary administrative (obtaining and studying the pick list) and set-up tasks (e.g. collecting a picking cart or vehicle),

- the time for traveling from the I/O point to the first storage location, between the storage locations, and back to the I/O point from the last storage location,

- the time needed at each storage location for identifying the storage location and the article, picking the required quantity, placing the items on the picking cart or vehicle, sorting them if necessary, and confirming the pick on the pick list, and

- the time needed at the I/O point for performing the final administrative and other closing tasks (unloading the picking cart or vehicle, sorting the collected items, depositing the transportation device, etc.).

Most of these time elements are independent from the above-described decisions. The exception is the time the order picker spends for traveling between the storage locations and from and to the I/O point. Thus, the minimization of the *total travel time*, i.e. the time necessary to collect all items of a given set of customer orders, can serve as a goal in decisions, which have to be made on the policy and operations levels.

Furthermore, it is usually realistic to assume (Jarvis/McDowell 1991, p. 94) that the total travel time is a monotone increasing function of the total distance the order pickers have to cover in order to collect all items of a set of customer orders (*total travel distance).* If the total travel distance is reduced, then also both the total travel time and the total picking time will decrease. Consequently, models and methods for decision problems on the policy and operations level often include minimization of the total travel distance as an auxiliary goal.

Finally, it has to be noted that – as long as the set of customer orders is fixed – it also makes sense to replace the goals introduced above by corresponding averages, i.e. to consider the minimization of the average total picking time per order, the average total travel time per order, or the average total travel distance (also: average tour length) per order instead.

4. Item Location

4.1. Dedicated Storage

4.1.1. Problem Definition

The item-location problem for dedicated storage can be stated as follows:

> Given an order-picking warehouse with specified potential locations for the storage of item types and a set of customer orders each requiring to collect a specific set of items, how should the item types be assigned to the available locations such that the total length of the picking tours necessary to collect all items is minimized.

It has to be noted that the item-location problem is strongly interconnected with both the order-batching and the picker-routing problem. In fact, only a simultaneous solution of all these problems could prepare the road to a "globally" optimal solution. Again, this is not a very realistic approach. Instead, one usually ignores the order-batching problem at this stage by simply taking the customer orders as picking orders. The remaining problem, however, still is a very difficult one. If we assume for a moment that we have solved the sequencing problem for each of the picking orders beforehand, then we would know how many times we have to proceed from an item type i to an item type j, and the remaining item-location problem could be modelled as a quadratic assignment problem. This optimization problem is known to be NP-hard, thus we conclude that the original item-location problem is also NP-hard at least. As the size of the instances of the quadratic assignment problem, which can be solved to an optimum, is rather limited (25 item types/locations) while – in relation to this size – real-world item-location problems tend to be quite large (more than 500 item types/locations), it cannot be expected that exact solution methods are applicable to real-world problem instances.

4.1.2. (Heuristic) Solution Approaches

Due to the fact that the item-location problem is NP-hard while the size of real-world problem instances of this problem type tends to be large, all methods suggested for the item-location problem in the literature so far are heuristics. They can be distinguished into three classes of methods, turnover-based, complementarity-based, and contact-based methods.

4.1.2.1. Turnover-Based Methods

Turnover-based methods (also: *volume-based* or *frequency-based methods*) consider the demand frequencies of the item types, only. Here, the turnover or demand frequency of an item type is equal to the number of times it appears in a customer order during a specific period of time. It can be expected that the distribution of these frequencies has a direct influence on the extent to which alternative storage policies affect picking times and costs (Kallina/Lynn 1976, Malmborg 1996). The distribution is usually represented by an ABC curve, which – in this context – can be characterized by a percentage ratio such as 20/80, which means that 20 % of the item types are responsible for 80 % of the demand frequencies (for details see Malmborg 1996, pp. 365 f.).

In case all item types require a single storage location each (i.e. the same amount of storage space), the item types are sorted in a non-increasing order of these frequencies, while, on the other hand, the available locations are sorted in a non-decreasing order with respect to their distances from the I/O point. Then the first item type is assigned to the first location, the second item is assigned to the second location, etc., yielding a solution in which frequently demanded items are located in close neighbourhood to the I/O point while less demanded items are located further away (for this *standard frequency-based location strategy* see Neal 1962). It can be shown (Kallina 1976) that this procedure provides an optimal solution for a rather restricted planning situation, namely when each picking order consists of one position each (i.e. in the case of an out-and-back system, which is more typical for a product-to-picker system). In the system considered here, an – often large – number of items has to be collected on a tour through the warehouse. As the standard frequency-based location strategy – like the other frequency-based methods which will be discussed below – ignores this aspect it cannot be expected that their application provides optimal (or even near-optimal) solutions for the above-stated item-location problem.

Instead of considering the exact distances between the available locations and the I/O point, in the literature, at times, "*storage-allocation patterns*" are used according to which item types are assigned to locations (Petersen 1999; Petersen/Schmenner 1999, pp. 487 ff.). Such storage-allocation patterns are depicted in Fig. 3. They indicate, where high-frequency, medium-frequency, and low-frequency item types are to be stored. Jarvis and McDowell (1991) have proven that when the traversal strategy is applied for solving the routing problem, within-aisle storage provides an optimal allocation scheme for symmetric order-picking warehouses with respect to the average tour length. They have also shown, however, that this is not necessarily the case for a non-symmetric warehouse, where the I/O point is not located in the middle of the front of the warehouse. Petersen and Schmenner (1999), by means of extensive numerical experiments, have evaluated – among other policies – diagonal, within-aisle, and across-aisle storage. They found that within-aisle storage generally outperforms diagonal and across-aisle storage, providing average tour length savings of 10-20 % (Petersen/Schmenner 1999, p. 494, p. 498).

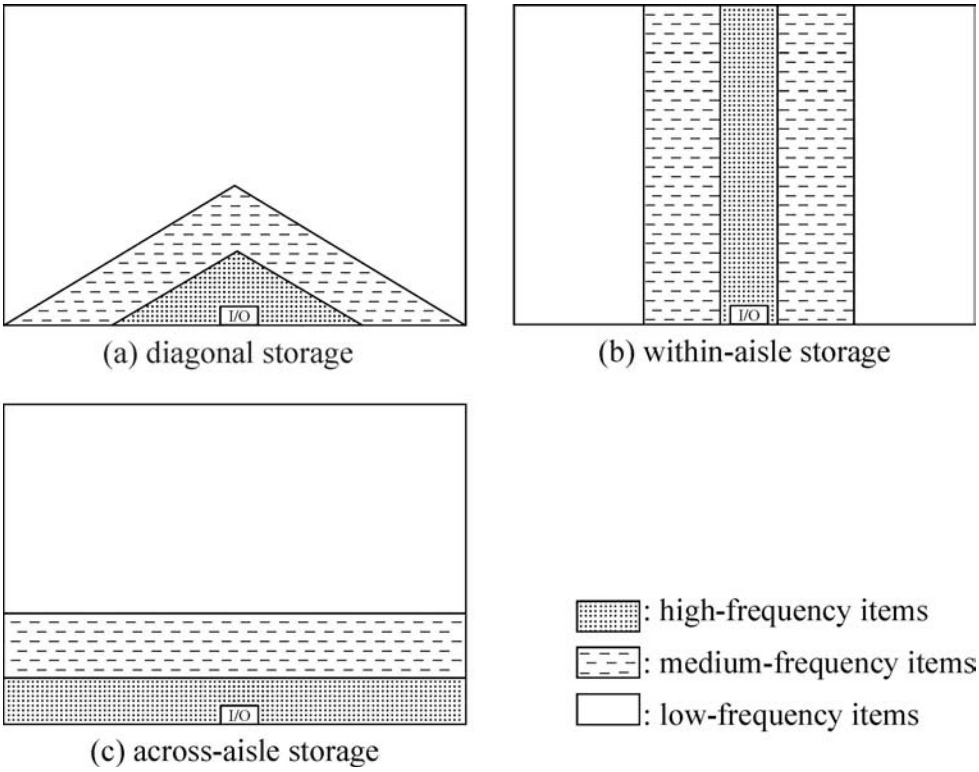


Fig. 3: Storage allocation patterns

An early modification of the standard frequency-based location strategy does not only take into account the demand frequencies of the item types but also their space requirements, which may be different for different item types. Heskett (1963, 1964) introduced the so-called *cube-per-order index* (COI), which is defined as the ratio of the required storage space to be allocated to an item type to its demand frequency per period (also see Ballou 1967, Kallina/Lynn 1976). The concept of the COI is to locate compact, high-frequency item types (low COI) close to the input-output point, and shift bulky, slow-moving items to remote storage locations. Several extensions of the COI concept have been described in the literature, in which aspects such as inventory cost (Wilson 1977), or zoning constraints (Malmborg 1995) have been considered. Again, results stemming from this research are of little value for the problem under discussion here, because they have been deduced under the assumption of a simple out-and-back system.

4.1.2.2. Complementarity-Based Methods

A second class of methods tries to take into account the complementarity between items, which is documented by the fact that certain pairs of item types are demanded together more frequently and therefore appear more often together in the same customer/picking order than others. These methods, which will be called *complementarity-based* here, cluster the item types into groups according to a measure of the strength of the joint demand (complementarity) and locate the members of a cluster as close to each other as possible. Thus, two major phases can be distinguished:

- identification of clusters of item types, which are demanded together frequently (*clustering problem*);
- clusterwise assignment of items to locations (*location-assignment problem*).

During the first phase, it has to be determined how the complementarity between the item types should be measured, how many clusters should be formed, and, of course, which item types go into each cluster.

Rosenwein (1994) has shown that the clustering problem can be formulated as a p-median problem, which is also frequently used in general cluster analysis (for a formulation of the p-median problem and its relationship with general cluster analysis see Mulvey/Crowder 1979). The (non-) complementarity (“distance”) of each pair (i, j) of item types i and j is defined by (Rosenwein 1994, p. 101)

$$w(i,j) = \sum_{q \in Q} | v(i,q) - v(j,q) | .$$

v(i,q) is a binary variable, which indicates whether item type i is included in customer order q:

$$(1) \qquad v(i,q) = \begin{cases} 1, & \text{if item type i is included in customer order q, } q \in Q, \\ 0 & \text{otherwise} \end{cases}$$

(Q: index set of all customer orders). A small value of w(i,j) indicates that item types i and j are frequently ordered together. By means of the p-median problem all item types are assigned to p clusters and from each cluster one item type is selected as a median, such that the sum of the non-complementarities between all item types and their respective median is minimized. Rosenwein (1994, p. 102) also introduces an approximation of the total travel distance, which is dependent on the number p of clusters. The number p*, which minimizes this approximation, represents the optimal number of clusters.

Liu (1999, p. 991) introduces a different complementarity measure, namely the similarity z(i,j) of a pair (i,j) of two item types i and j:

$$(2) \qquad z(i,j) = \frac{1}{|Q|} \sum_{q \in Q} \frac{\min[u(i,q), u(j,q)]}{\max[u(i,q), u(j,q)]} \quad ,$$

where u(i,q) is the demand (i.e. the number of units) required of item type i in customer order q. z(i,j) can be interpreted as the probability that i and j appear together in the same order. For z(i,j) = 1, item types i and j are always ordered together, while for z(i,j) = 0, item types i and j never appear together in the same order. Liu also uses the p-median problem to cluster the item types (however, due to the fact that a similarity measure instead of a distance measure has been used, he maximizes the sum of distances between all item types and their respective median), but he does not indicate how the number of clusters can be determined.

Rosenwein (1994) does not explain how the subsequent problem of assigning the clustered item types to locations should be solved. Liu (1999, pp. 997 f.) proposes to identify the item type with the largest demand and assign it to a location closest to the I/O point. Then all other item types of the same cluster are assigned to locations according to the standard frequency-based location strategy. These steps are repeated until all item types have been assigned. Liu exemplifies his approach on a simple order-picking system, which basically consists of one picking aisle. However, for a multi-aisle system, location assignment will have to follow one of the storage-allocation patterns (diagonal, within-aisle, and across-aisle storage) described above. Solutions obtained in this way appear not be very satisfactory, as it is very likely – especially with a small number of clusters – that item types of a single cluster have to be split between different aisles. That, however, contradicts the original idea of the complementarity-based approach of locating jointly demanded item types in close proximity.

4.1.2.3. Contact-Based Methods

Another basic drawback of the complementarity-based approach to the item-location problem can be seen in the fact that, even though two item types i and j appear in the same customer order, an order picker will not necessarily proceed directly from the location of item type i to the location of item type j (or vice versa). Thus, in order to minimize the total travel distance, it would be more desirable to take into account the number of times an order picker really travels directly between the locations of two item types i and j, i.e. the number of times she/he picks either item type j directly after item type i, or item type i directly after item type j, respectively. Methods, which allocate item types to locations with respect to the number of these direct travels (*contacts*), will be called *contact-based*, here. Unfortunately, the *contact frequencies* c(i,j) between item types i and j are not known in advance but stem from the (optimal) solutions of the routing problems related to the set of customer/picking orders under consideration. The solution of the routing problems, however, is dependent on the location of the item types, which again demonstrates the strong interrelationship between item location and routing.

The method of van Oudheusden, Tzen, and Ko (1988, pp. 279 f.) can be characterized as a classic local-search method. It starts from an initial allocation of item types to storage locations, for which the distances d(i,j) between all pairs (i,j) of item types i and j are computed.

Given these distances, an (optimal) solution is determined for each of the routing problems associated with the set of customer/picking orders. These solutions provide the contact frequencies c(i,j) for every item pair (i,j). The total travel distance TTD associated with the initial item allocation can be calculated as

$$(3) \qquad \mathrm{TTD} = \sum_{i\in I} \sum_{j\in I} d(i,j)\cdot c(i,j)$$

(*I*: index set of item types). Based on the current values of the contact frequencies c(i,j), the method tries to improve the value of (3) by means of a pairwise exchange of the locations of two item types. The improvement phase is performed in terms of a strict hill-descending procedure in which an exchange of locations is accepted only if the value of (3) is reduced. After each exchange, the distances d(i,j) are updated, while the contact frequencies c(i,j) are left unchanged for the time being. This is for the obvious reason that an update of the c(i,j)-values is a very time-consuming activity because it requires to determine all picking tours anew. The contact frequencies c(i,j) will only be updated when no further exchange of locations can be identified, which improves the value of (3). At that stage, the routing problems are resolved and the new, “real” value of TTD is computed on the basis of the new contact frequencies. The current item allocation now serves as the starting point for another execution of the improvement phase. The method terminates if the contact frequencies have been updated but no pairwise exchange of item locations can be found that would improve the current objective function value (3). The authors applied their approach to real-world data from a central warehouse of an integrated steel mill. They found that the existing, obviously not very well planned item allocation could be improved significantly and that, in connection with improved routing, savings in picker travel time of up to 83 % were possible (van Oudheusden/Tzen/Ko 1988, pp. 281 f.).

Reschke and Wäscher (2000) present three local search-methods, which are also based on the pairwise exchange of storage locations. However, their methods are different to that of van Oudheusden, Tzen, and Ko in the sense that the exact value of the new total travel distance TTD is computed each time before the decision is made about whether to accept an exchange or not. Reschke and Wäscher demonstrate that it is not necessary to determine a new (optimal) tour for every customer/picking order at each iteration but that it is sufficient to consider only such tours, which contain exactly one of the two exchange items (Reschke/Wäscher 2000, pp. 143 f.). The first proposed method is a classic (2-opt) hill-descending method, which only accepts an exchange of locations if it improves the current solution. The remaining two methods are based on Simulated Annealing (e.g. see van Laarhoven/Aarts 1992) and Threshold Accepting (see Dueck/Scheuer 1990), which – at least temporarily – allow the objective function value (3) to deteriorate in order to avoid being trapped in a local minimum.

4.2. Randomized Storage

In randomized storage a dispatching rule has to be chosen according to which an incoming pallet or an incoming lot of a particular item type is assigned to one (or several, if necessary) of the currently available, empty locations. Basically, two dispatching rules have been proposed in the literature (Hausman/Schwarz/Graves 1976, p. 631).

- *Completely random storage assignment*: An incoming pallet is assigned to an empty location completely at random; the probability that an incoming pallet is assigned to a particular location is the same for all empty locations.

- *Closest open location*: An incoming pallet is assigned to a location among the available locations, which lies closest to the I/O point; the pallet is stored at this location, regardless of the demand frequency of its item type.

Hausman, Schwarz, and Graves (1976, p. 631) have shown, that both rules are equivalent in the sense, that – given a constant number of used bays – in steady state all stores are equally likely to take place at any of these bays.

4.3. Discussion

4.3.1. Dedicated vs. Random Storage

When dedicated and randomized storage are compared in the literature, usually a trade-off between retrieval efficiency and space requirements is postulated. Dedicated storage appears to be more efficient in terms of the time, which is needed to retrieve items from the warehouse while randomized storage tends to minimize storage space requirements (Malmborg 1996, pp. 363 f.). Dedicated storage requires extensive storage space because, for example, the reservation of a location for a certain item type has to be maintained even if this article is temporarily out of stock. On the other hand, with randomized storage whenever a storage location has become vacant, it can be allocated immediately to an incoming pallet. However, the space-argument in favour of dedicated storage is only a valid one as long as design issues (such as warehouse size and layout) are addressed. For a given order-picking warehouse – which is the situation considered here – usually the available space has to be looked upon as fixed. Therefore, the corresponding "space cost" (cost of depreciation in particular) are also fixed, and the minimization of space requirements does not constitute a relevant goal.

Furthermore, it has been argued (Malmborg 1996, p. 363) that randomized storage provides a more compact allocation of space. Items are located closer to each other, resulting in shorter picking tours than in dedicated storage where the items to be picked are located further apart. Malmborg (1996, pp. 373 f., p. 376) has indeed found that lower average retrieval costs can be obtained by randomized storage rather than by dedicated storage when the level of variation in the retrieval frequencies is low. He determined the critical value of the variation to be given by a 10/33-ABC curve. For situations with more skew (which is very likely to be the case in real-world order-picking warehouses) tour-length reductions from compact space allocation are offset by an increased number of travels to distantly located, high-frequency item types.

Petersen (1999) carried out numerical experiments in which he compared within-aisle and diagonal storage to randomized storage. The performance measure was the total picking time per picking order. He reported (Petersen 1999, pp. 1060 f.) that within-aisle storage was between 30 % and 9 %, diagonal storage between 25 % and 6 % better than random storage, where the difference in performance decreased with an increasing number of positions per picking order.

Therefore, with respect to the original goal of total picking-time minimization, it has to be concluded that for an order-picking warehouse of the standard type, dedicated storage is to be preferred to randomized storage. This is also confirmed by the fact that – due to having fixed the locations of the various item types for a relatively long period of time – order pickers become familiar with the location of items, which will reduce the actual picking time even further.

4.3.2. Performance of Heuristic Solution Methods for the Item-Location Problem

Even though dedicated storage appears to be more suitable for an order-picking warehouse, which of the above-introduced methods is to be preferred for solving the item-location problem is still a question that has not been answered in a completely satisfactory manner, so far. It has to be stated, that there is very little reliable information available with respect to the (relative) solution quality and computing times of the presented algorithms.

According to the best of our knowledge, the numerical experiments carried out by Reschke and Wäscher (2000, pp. 144 ff.) are the only ones in which several methods have been directly compared to each other so far. The focus of these experiments was on contact-based methods. The authors implemented the method by van Oudheusden, Tzen, and Ko and the three above-described local search-methods and applied them to 140 randomly generated problem instances. The problem instances were grouped into 14 problem classes characterized by parameters such as the average number of positions per customer/picking order, degree of complementarity between item types and skew of the ABC-curve of the demand frequencies. Initial solutions were provided by the standard frequency-based location strategy. The corresponding objective function values also served as benchmarks for the assessment of the solution quality.

It could be shown that all local search-methods improve the initial solutions significantly. The solution quality of Simulated Annealing and of Threshold Accepting turned out to be almost equivalent. On the average, these methods reduced the objective function value to 61-77 % of the benchmark. Generally, the results indicate the existence of significant opportunities for picker-productivity improvement in practice, where items are usually assigned to storage locations by means of the standard frequency-based location strategy. On each of the problem classes, Simulated Annealing and Threshold Accepting both outperformed the classic 2-opt hill-descending method, which, on average, reduced the objective function values down to 66-77 % of the benchmark. The hill-descending method, in turn, outperformed the method by van Oudheusden, Tzen, and Ko, which achieved a reduction down to 69-83 % of the benchmark. For Simulated Annealing and Threshold Accepting the improvements were generally larger on problem instances with a smaller number of picks per customer/picking order, less skew of the ABC frequency curve and a smaller degree of complementarity.

On the other hand, improvements in the solution quality come at the expense of dramatically increased computing times. As could be expected with respect to the extensive computations necessary at each iteration (determination of the new optimal picking tours), the computing times per problem instance can easily amount to hours for Simulated Annealing and Threshold Accepting on a PC. However, such computing times may still be acceptable in practice, because the item-location problem is usually not a time-critical one and solved in long time intervals (once a year or once in six months) only.

5. Order Batching

5.1. Problem Statement

The order-batching problem can be stated as follows:

> Given a set of customer orders, each consisting of a number of orderlines, a given assignment of items to storage locations in a standard order-picking warehouse and a given capacity of the picking device, how should the customer orders be grouped (batched) into picking orders such that no customer order is split between two or more picking orders and the total lengths of all picker tours necessary to collect all items is minimized.

The batching problem is also known to be NP-hard. Consequently, with respect to the size of problem instances encountered in practice, research has concentrated on methods of the heuristic type.

5.2. Solution Methods

The heuristics suggested in the literature for the order-batching problem can be classified into *priority rule-based algorithms, seed algorithms and savings algorithms.*

5.2.1. Priority Rule-Based Algorithms

These algorithms (see Fig. 4) assign a priority to each customer order, at first. Then, in the order given by the priorities, the customer orders are assigned one by one to batches (picking orders) in a way that the capacity constraint is not violated.

Several suggestions have been made in the literature with respect to the determination of the priorities (Rule PR1). The most straightforward specification is the First-Come-First-Serve (FCFS) Rule. Gibson and Sharp (1992) suggested two-dimensional and four-dimensional space-filling curves to be used for the determination of the priorities. These approaches map the coordinates of the locations of the items of a customer order into a (theta-) value on the unit circle. The theta-values range from 0 to 1. The larger the theta-value is, which has been assigned to a customer order, the higher is its priority.

Next-Fit (batches are completed with orders in the sequence given by the priorities; every time the addition of another customer order would violate the capacity constraint, a new batch is started), First-Fit (batches are numbered in the sequence in which they are started; the current customer order is assigned to a batch with the smallest number into which it fits), and Best-Fit (of those batches into which a customer order would fit, it is assigned to that one where it leaves the smallest remaining capacity) are specifications of the *batch selection rule* (Rule PR2).

```
Procedure

assign a priority (PR1) to each customer order;
sort all customer orders according to non-ascending priorities;
open first batch and assign first order to it;
while customer orders unassigned do
    select next customer order;
    if customer order can be assigned to an open batch then
        select open batch (PR2);
        assign customer order to selected batch;
    else
        open new batch;
        assign customer order to new batch;
    endif
endwhile
```

Fig. 4: Pseudocode for priority rule-based algorithms

5.2.2. Seed Algorithms

Seed Algorithms have been introduced into the literature by Elsayed (1981) and by Elsayed and Stern (1983), who considered them for automatic warehouses in the first place. A systematic study for manual warehouses is presented in de Koster, van der Poort, and Wolters (1999). Seed methods generate batches sequentially, i.e. a new batch is not started before the current one has been closed. In order to form a batch, a customer order is selected as the so-called "seed" of the batch. Further, not yet assigned customer orders are added to the batch until its capacity is exhausted. Fig. 5 gives a more formal description of the procedure.

For the *seed-selection rule* (Rule SR1), several options have been discussed in the literature (de Koster/van der Poort/Wolters 1999, p. 1483), namely – among others – the selection of a random order, an order with the largest number of positions (i.e. selection of the "largest" order), an order with the longest picking tour, an order with the most distantly-located item (i.e. the item located furthest away from the depot), an order with the largest number of aisles to be visited, an order with the largest aisle range (i.e. the largest difference between the aisle numbers of the right-most and the left-most aisle to be visited), etc. The seed-selection rule can be applied in two ways (de Koster/van der Poort/Wolters 1999, p. 1484), in a *single mode* (in which the originally selected customer order only serves as seed for the present batch) or a *cumulative mode* (in which all customer orders already assigned to the current batch make up for the seed of the batch).

```
Procedure

while customer orders unassigned do
    open new batch;
    select an unassigned customer order as seed order (SR1);
    while customer orders exist which can be assigned to open batch do
        select an unassigned customer order (SR2)
            which does not exceed the remaining capacity of the present batch;
        assign selected customer order to open batch;
    endwhile
    close current batch;
endwhile
```

Fig. 5: Pseudocode for seed algorithms

The *order-congruency rule* (Rule SR2) determines which unassigned customer order should be the next one to be added to the current batch. Usually, an order is selected which "distance" to the seed of the current batch is minimal. This distance between an unassigned order and the seed can be defined in several ways, e.g. (de Koster/van der Poort/Wolters 1999, p. 1484) as

- the sum of the travel distances (measured in length or time units or in the number of aisles) between every location of a seed item and the closest location of any item in the order,
- the sum of the travel distances between every location of an item in the order and the closest location of any item in the seed,
- the number of additional aisles, which have to be visited if the order would be added to the seed,
- the difference between the gravity centre of the seed and the gravity centre of the order (where the gravity centre of an order is defined as the average aisle number of the locations of the items in an order), etc.

5.2.3. Savings Algorithms

Savings algorithms (de Koster/van der Poort/Wolters 1999, pp. 1485-1487; Elsayed/Unal 1989, pp. 1099-1101) are based on the well-known Clarke-and-Wright-Algorithm for the vehicle-routing problem (Clarke/Wright 1964). Let $d(q)$ be the length of the picking tour for order q, $d(r)$ the length of the respective tour for order r, and $d(q \circ r)$ the length of a single picking tour in which the items of both order q and r are collected. Thus, the savings $s(q,r)$ of combining the two orders q and r in a single tour can be computed as $s(q,r) = d(q) + d(r) - d(q \circ r)$. The structure of the basic algorithm is depicted in Fig. 6.

A recalculation of the savings s(q,r) any time one or two customer orders have been assigned to a batch is likely to improve the solution quality of the algorithm.

The EQUAL-Algorithm (Elsayed/Unal 1989, pp. 1099 f.) is a savings algorithm that generates batches sequentially. The first batch is started with a pair of customer orders q and r, which can be combined in a batch and for which the savings s(q,r) are maximal. These two orders are taken as an initial seed. Then an order is added to the batch, which is chosen such that the savings of combining the seed with an unassigned order is maximized and the capacity of the picking device is not violated. The old seed and the newly added customer order form the new seed. This is repeated until the capacity of the picking device does not permit to add another customer order. Then a new batch is opened and completed in the same way etc. In the end, customer orders might be left, which cannot be combined with each other, due to the limited capacity of the picking device. They are assigned to an individual batch each.

```
Procedure

determine the savings s(q,r) for all pairs (q,r) of customer orders q and r, which can be combined with respect to the given capacity of the picking device;
sort the order pairs in non-ascending order;
while at least one pair of customer orders unchecked do
    select next pair (q,r) of customer orders q and r;
    if both order q and order r have already been assigned then
        select next pair (q,r) of customer orders q and r;
    endif
    if neither customer order q nor order r have been assigned previously then
        open a new batch;
        assign both order q and order r to the new batch;
    endif
    identify order q*, q* ∈ {q,r} from the pair (q,r) of customer orders q and r, which has already been assigned to a batch;
    identify the batch k(q*) to which q* has been assigned;
    if the unassigned order r* = {q,r} \ q* can be assigned to k(q*) then
        assign r* to k(q*);
    endif
endwhile
assign each unassigned customer order to an individual batch.
```

Fig. 6: Pseudocode for savings algorithms

In the Small-Large- (SL-) Algorithm (Elsayed/Unal 1989, p. 1100), the set of customer orders is divided into two subsets, namely into a set of large orders (containing a number of items larger than a pre-specified number) and into a set of small orders (containing the remaining orders). To the set of large orders, the EQUAL-Algorithm is applied. Next, the set of small orders is sorted in non-ascending order of their size. In this sequence, the small orders are assigned, each order to that batch where it results in the largest savings without violating the remaining capacity. For an order, which cannot be assigned, a new batch is opened.

5.3. Discussion

The FCFS-Heuristic cannot really be considered as a competitive method for the solution of the order-batching problem. In fact, it is usually only taken as a benchmark method to which other methods are compared in terms of solution quality and computing times. Gibson and Sharp (1992) have carried out numerical experiments with priority rule-based algorithms, in which the priorities were determined by means of space-filling curves, and with a basic seed algorithm, which they introduced into the literature under the name of Sequential Minimum Distance (SMD) Batching Heuristic. They found (Gibson/Sharp 1992, p. 67) that for warehouses where the distances between locations are measured according to an aisle metric (and not in terms of a Euclidian, Chebyshev, or rectilinear metric), the SMD Heuristic provided the best solutions.

De Koster, van der Poort, and Wolters (1999) have carried out extensive numerical experiments with seed and savings algorithms. The heuristics were evaluated with respect to different parameters such as the warehouse type, the location of the depot, and the number of orders per batch. Two routing strategies, traversal and largest-gap, were considered, random vs. class-based storage investigated. The total travel time served as the overall performance measure, permitting that influencing factors such as the travel speed within aisles, the speed outside aisles, and the time to enter and leave aisles could be considered. The authors concluded (de Koster/van der Poort/Wolters 1999, pp. 1498 f.) that the routing strategy should be selected first before the batching strategy is chosen. In case of a large number of items to be picked per aisle and substantial aisle-changing time, the traversal strategy is favourable, otherwise the largest-gap strategy is to be preferred. If the traversal strategy is used and the capacity of the picking vehicle is large, the traversal strategy provides the best results. On the other hand, if the largest-gap strategy is applied and the capacity of the picking vehicle is small, then savings algorithms provide superior results. In this situation, the best results were obtained by the basic algorithm, which included recalculation of the savings s(q,r). However, these results came at the expense of long computing times. It consumed about ten times the computing time of the other savings algorithms and about 100-200 times the computing time of seed algorithms.

6. Picker Routing

6.1. Problem Statement

The third problem to be considered here concerns the routing of the order pickers. It can be formulated as follows:

> Given an assignment of items to storage locations in a standard order-picking warehouse and a set of picking orders each requiring to pick a certain set of items, in which sequence should the locations of the items in each picking order be visited such that the length of the total travel distance is minimized.

We note that, at this stage, the picking orders are fixed. The total length of all picking tours will be minimized if each tour length is minimized individually. Thus, for each picking order, a Traveling Salesman Problem (TSP) has to be solved to give an optimal tour between the locations of the items to be picked. The general TSP is known to be NP-hard. However, Ratliff and Rosenthal (1983) have shown that the above TSP related to order picking in a rectangular warehouse can be solved in polynomial time.

6.2. Solution Methods

6.2.1. Exact (Optimal) Methods

In order to formulate a model of the picker-routing problem, an approach outlined in de Koster/van der Poort (1998), Roodbergen (2001), and Roodbergen/de Koster (2001b) will be described. According to this approach, the picker-routing problem in a standard warehouse can be modelled by means of a valued, undirected multi-graph G. Let m denote the number of locations from which items have to be picked as defined by the corresponding picking order. Furthermore, let n denote the number of picking aisles in the warehouse. Then the vertex set *V* of the graph G consists of m elements v_i, each one representing a pick location i (i = 1, …, m), n elements a_j, where a_j represents the location of the front end of aisle j, n elements b_j, where b_j represents the location of the rear end of aisle j (j = 1, …, n), and an element v_0 representing the location of the I/O point.

Any two vertices which correspond to adjacent locations in the warehouse are connected by a pair of undirected, parallel arcs. Two arcs are introduced because the path between two adjacent locations may be chosen more than once in a picking tour, while – on the other hand – in an optimal (minimum length) tour the order picker will not have to walk more than twice between any pair of locations (Ratliff/Rosenthal 1983, p. 510). The values assigned to each arc represent the length of the path between the corresponding two locations. A multi-graph G, which has been constructed in this way to represent a routing problem in a standard warehouse, will be called an *order-picking graph* (de Koster/van der Poort 1998, p. 471). For the routing problem presented in Fig. 2a, Fig. 7 depicts the corresponding order-picking (multi-) graph.

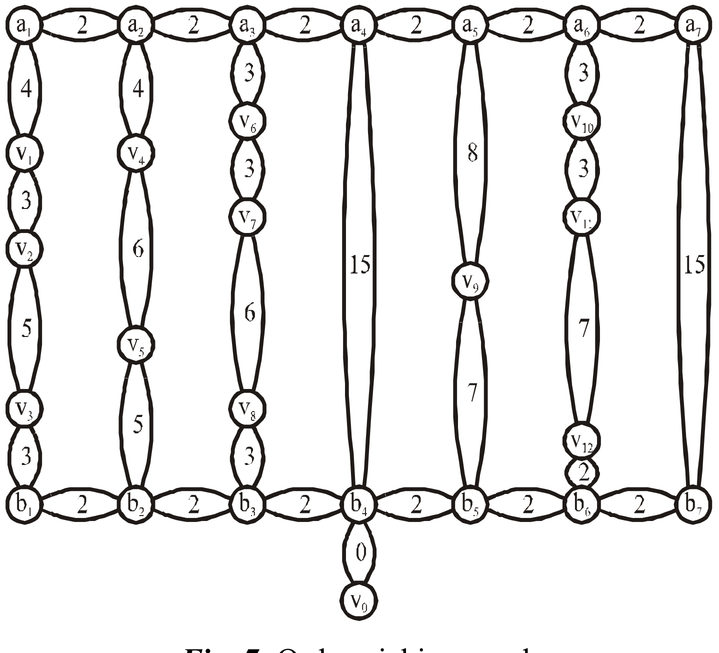


Fig. 7: Order-picking graph

An order-picking tour is a cycle in G that contains each of the vertices v_i (i = 0, 1, …, m) at least once, the length of the tour is the sum of the values of the arcs included in the cycle. A subgraph T of G that contains all vertices v_i (i = 0, 1, …, m) of G will be called a *tour subgraph* of G if there is an order-picking tour that uses each arc in T exactly once. As an order-picking tour can be generated from a given tour subgraph by a very simple (efficient) procedure (Ratliff/Rosenthal 1983, pp. 518 ff.), the problem of constructing a minimum-length picking tour in an order-picking graph of a standard order-picking warehouse is reduced to finding a minimum-length tour subgraph of the corresponding order-picking graph. Ratliff and Rosenthal (1983, pp. 513-516) present an algorithm for this problem, which is linear in the number of aisles. Therefore, their optimal algorithm is fast enough to be applied to any real-world instance of the picker-routing problem in standard warehouses. Fig. 8 presents a (minimum-length) tour subgraph of the order-picking graph shown in Fig. 7, and Fig. 9 the corresponding optimal tour.

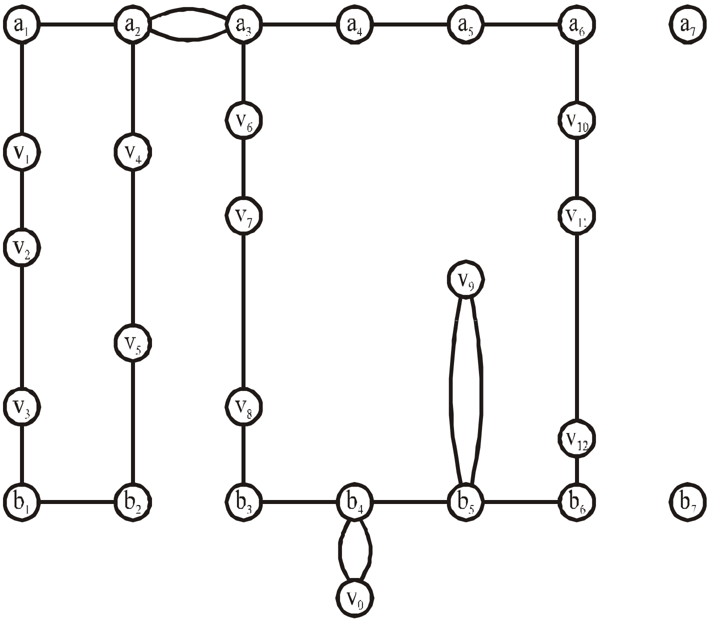


Fig. 8: Minimum-length tour subgraph

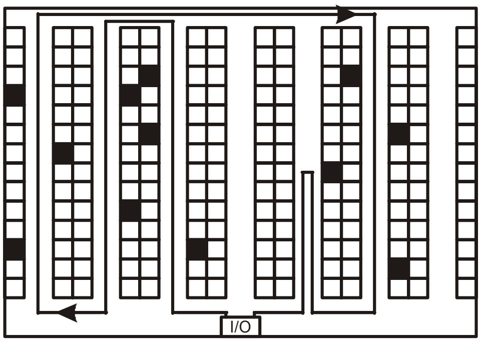


Fig. 9: Optimal tour corresponding to the tour subgraph of Fig. 8

Goetschalckx and Ratliff (1988) developed an efficient optimal algorithm for the picker-routing problem in a one-block warehouse with wide aisles, where the order picker cannot reach items on both sides without changing her/his position. Furthermore, the authors discovered that unless the pick density (i.e. the number of the locations from which items have to be picked in relation to the total number of locations) is greater than 50 %, picking from both sides of the aisle in the same pass yields significantly shorter tours than picking one side first and then returning on the other side.

Based on the algorithm by Ratliff and Rosenthal, several approaches have been developed for solving picker-routing problems in other, non-standard warehouses. In de Koster/van der Poort (1998) the algorithm is modified for solving such problems in warehouses with decentralized depositing, in Roodbergen/de Koster (2001b) for problems in order-picking warehouses with a middle cross aisle. In Roodbergen/de Koster (2001a) the authors use a branch-and-bound method to obtain optimal tours for warehouses with multiple cross aisles.

6.2.2. Heuristic Methods

The above-introduced routing strategies (traversal, return, largest-gap, composite) can be seen as heuristic solution methods for the picker-routing problem. Initially, research efforts concerning this type of methods concentrated on the development of approximations for the (average) tour length. Kunder and Gudehus (1975) consider the traversal and the return strategy and present formulae from which the (expected) average travel time per pick can be computed for any value of parameters such as the number of aisles, aisle length, aisle width, number of articles in the warehouse, number of pick locations, picker speed, etc. Their research is extended by Hall (1993), who considers the traversal and largest-gap strategy. His analysis shows that the largest-gap strategy is to be favoured in warehouses with narrow aisles if the number of picks is less than 3.8, otherwise the traversal strategy appears to be superior. Petersen (1997) compared – among others – traversal, return, largest-gap, and composite routing strategies by means of numerical experiments. He found that the composite and largest-gap heuristics perform best. In these three papers, however, analysis and experiments are carried out under the assumption that pick locations are independent and uniformly distributed (which does not appear to be very realistic for order-picking warehouses).

Caron, Marchet, and Perego (1998) investigate the expected total travel distance of the traversal and the return strategy for COI-based storage. The deduced approximations are validated by means of numerical experiments. The authors find that whether the traversal or the return strategy has to be preferred is affected by the average number of picks per aisle and the skew of the ABC frequency curve in the first place while the number of aisles is a less critical parameter. The return strategy outperforms the traversal strategy only for a small number of picks per aisle (i.e. < 1) and for skewed ABC curves; otherwise the traversal strategy is superior. For a large number of picks per tour the traversal strategy outperforms the return strategy, irrespective of the skew of the ABC curve (Caron/Marchet/Perego 1998, p. 731).

Petersen (1999) compares several routing strategies (traversal, largest-gap, composite) in a volume-based storage environment by means of extensive numerical experiments. The performance measure is the (average) total time necessary to collect all items of a picking order and includes a time-estimate for all non travel-related activities. His findings are (Petersen 1999, pp. 1059 f.) that for within-aisle storage the composite routing strategy performs best. For small picking orders the performance of the largest-gap strategy is almost identical, but it becomes inferior for picking orders containing more than nine positions. For order sizes of more than 28 positions the largest-gap strategy even becomes inferior to the traversal strategy. For diagonal storage, both composite and largest-gap strategies perform almost equally well and outperform the traversal strategy for all order sizes considered (up to 50 positions).

6.3. Discussion

The application of a routing heuristic instead of an optimal algorithm involves accepting an inferior solution quality. De Koster and van der Poort (1998) carried out a series of numerical experiments in order to compare the performance of the traversal heuristic to that of an optimal algorithm for three different types of order-picking warehouses (narrow-aisle high-bay warehouse, shelf picking, and standard warehouse). The authors assumed randomly distributed picking locations. For the first two types they found that the traversal strategy generated solutions, which were – on average – between 7.3 % and 20.8 % above the optimum in terms of travel time and between 2.4 % and 5.8 % above the optimum in terms of total picking time. The differences were more significant for the standard warehouse, in which the heuristic rated above the optimum between 26.6 % and 34.2 % with respect to travel time and between 11.2 % and 13.9 % with respect to total picking time. In Petersen's experiments (Petersen 1997) also random storage has been assumed. The heuristics which came out best, composite and largest-gap, were found to be 9-10 % over optimal on average in terms of tour length. In another series of experiments, in which frequency-based storage was assumed, Petersen and Schmenner (1999) compared the solutions of several heuristics to those obtained by an optimal algorithm. The more advanced heuristics, largest-gap, composite, and midpoint, came out, on average, 7.1 %, 10.2 %, and 12.0 % over optimal. With 27.9 % and 30.9 % over optimal, the simpler return and traversal strategies performed significantly worse (Petersen/Schmenner 1999, p. 492).

Given the fact that the Ratliff-Rosenthal-Algorithm solves the picker-routing problem in polynomial time and can be applied to any real-world problem instance, and given the inferior solution quality of heuristic routing methods, one may ask why anyone would use a heuristic method. First of all, our discussion has focussed on routing problems in standard order-picking warehouses. Real-world warehouses may have different layouts or other specific properties for which no exact algorithm has been developed so far. Secondly, the difference between a minimum-length tour and a heuristically-obtained solution may not always be really significant, because practical situations exist in which total travel time represents only a relatively small proportion of the total picking time. De Koster and van der Poort (1998, p. 478) mention that the travel time is often only about 50 % of the (total) picking time. Finally, optimal tours may not always work in practice, because they can be confusing for the order picker and increase the probability of missing picks. Heuristics, on the other hand, often can be understood and remembered more easily and require less concentration when being executed. These advantages of heuristics have to be compared against the savings in travel time and travel distance from optimal routing (Petersen 1999, p. 1054).

7. Conclusions

It should have become clear from the previous discussion that by means of advanced planning techniques, which are available today, it is possible to improve the performance of order-picking operations significantly, both with respect to order processing times and picking costs. On the other hand, it is also obvious that the present knowledge is still insufficient in several areas and that additional studies are necessary to improve the state-of-the-art. This concerns the solution methods for the item-location problem, which seem to provide good-quality solution but require too much computing effort. With respect to the order-batching problem it is striking that only construction methods have been presented in the literature so far, while neither any improvement methods have been suggested nor the application of meta-heuristics has been investigated. Finally, due to the close relationship between order-batching and optimal routing, it might be worthwhile to consider a simultaneous solution approach for these two problem areas.

Literature

Ballou, R. H. (1967):
Improving the Physical Layout of Merchandise in Warehouses. *Journal of Marketing* **31**, July, 60-64.

Bassan, Y.; Roll, Y.; Rosenblatt, M. J. (1980):
Internal Layout Design of a Warehouse. *AIIE Transactions* **12**, 317-322.

Caron, F.; Marchet, G.; Perego, A. (1998):
Routine Policies and COI-Based Storage Policies in Picker-to-Part Systems. *International Journal of Production Research* **36**, 713-732.

Caron, F.; Marchet, G.; Perego, A. (2000):
Optimal Layout in Low-Level Picker-to-Part Systems. *International Journal of Production Research* **38**, 101-117.

Clarke, G.; Wright, W. (1964):
Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* **12**, 568-581.

Cormier, G.; Gunn, E. A. (1992):
A Review of Warehouse Models. *European Journal of Operational Research* **58**, 3-13.

de Koster, M. B. M.; van der Poort, E. S.; Wolters, M. (1999):
Efficient Orderbatching Methods in Warehouses. *International Journal of Production Research* **37**, 1479-1504.

de Koster, R.; van der Poort, E. (1998):
Routing Orderpickers in a Warehouse: A Comparison Between Optimal and Heuristic Solutions. *IIE Transactions* **30**, 469-480.

Dueck, G.; Scheuer, T. (1990):
Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing. *Journal of Computational Physics* **90**, 161-175.

Elsayed, E. A. (1981):
Algorithms for Optimal Material Handling in Automatic Warehousing Systems. *International Journal of Production Research* **19**, 525-535.

Elsayed, E. A.; Stern, R. G. (1983):
Computerized Algorithms for Order Processing in Automated Warehousing Systems. *International Journal of Production Research* **21**, 579-586.

Elsayed, E. A.; Unal, O. I. (1989):
Order Batching Algorithms and Travel-Time Estimation for Automated Storage/Retrieval Systems. *International Journal of Production Research* **27**, 1097-1114.

Gibson, D. R.; Sharp, G. P. (1992):
Order Batching Procedures. *European Journal of Operational Research* **58**, 57-67.

Goetschalckx, M.; Ratliff, H. D. (1988):
Order Picking in an Aisle. *IIE Transactions* **20**, No. 1, 53-62.

Hall, R. H. (1993):
Distance Approximations for Routing Manual Pickers in a Warehouse. *IIE Transactions* **24**, No. 4 (July), 76-87.

Hausman, W. H.; Schwarz, L. B.; Graves, S. C. (1976):
Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science* **22**, 629-638.

Heskett, J. L. (1963):
Cube-per-Order Index – A Key to Warehouse Stock Location. *Transportation and Distribution Management* **3**, 27-31.

Heskett, J. L. (1964):
Putting the Cube-per-Order Index to Work in Warehouse Layout. *Transportation and Distribution Management* **4**, 23-30.

Jarvis, J. M.; McDowell, E. D. (1991):
Optimal Product Layout in an Order Picking Warehouse. *IIE Transactions* **23**, 93-102.

Kallina, C. (1976):
Optimality of the Cube-per-Order Index Rule for Stock Location in a Distribution Warehouse. Working Paper, American Can Company, March 1976.

Kallina, C.; Lynn, J. (1976):
Application of the Cube-per-Order Index Rule for Stock Location in a Distribution Warehouse. *Interfaces* **7**, No. 1 (November), 37-46.

Kunder, R.; Gudehus, T. (1975):
Mittlere Wegzeiten beim eindimensionalen Kommissionieren. *Zeitschrift für Operations Research* **19**, B53-B72.

Liu, C.-M. (1999):
Clustering Techniques for Stock Location and Order-Picking in a Distribution Center. *Computers & Operations Research* **26**, 989-1002.

Malmborg, C. J. (1995):
Optimiztion of Cube-per-Order Index Layouts with Zoning Constraints. *International Journal of Production Research* **33**, 465-482.

Malmborg, C. J. (1996):
Storage Assignment Policy Tradeoffs. *International Journal of Production Research* **34**, 363-378.

Mulvey, J. M.; Crowder, H. (1979):
Cluster Analysis: An Application of Langrangean Relaxation. *Management Science* **25**, 329-340.

Neal, F. L. (1962):
Controlling Warehouse Handling Costs by Means of Stock Location Audits. *Transportation and Distribution Management* **2**, May, 31-33.

Petersen II, C. G. (1997):
An Evaluation of Order Picking Routing Policies. *International Journal of Operations and Production Management* **17**, 1098-1111.

Petersen II, C. G. (1999):
The Impact of Routing and Storage Policies on Warehouse Efficiency. *International Journal of Operations and Production Management* **19**, 1053-1064.

Petersen II, C. G.; Schmenner, R. W. (1999):
An Evaluation of Routing and Volume-based Storage Policies in an Order Picking Operation. *Decision Sciences* **30**, 481-501.

Ratliff, H.; Rosenthal, A. (1983):
Orderpicking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research* **31**, 507-521.

Reschke, V.; Wäscher, G. (2000):
Local Search-Verfahren für die Lagerplatzvergabe in Mann-zur-Ware-Kommissioniersystemen. *Logistik 2000plus: Herausforderungen, Trends, Konzepte. Begleitband zur 6. Magdeburger Logistik-Tagung, Magdeburg, 16./17.11.2000* (Hrsg.: Inderfurth, K.; Schenk, M.; Ziems, D.). Magdeburg: Otto-von-Guericke-Universität und LOGiSCH GmbH, 136-151.

Roodbergen, K. J. (2001):
Layout and Routing Methods for Warehouses. Doctoral Dissertation, Erasmus Research Institute of Management (ERIM), Erasmus University, Rotterdam (ERIM Ph.D. Series Research in Management 4, TRAIL Thesis Series No. T2001/3).

Roodbergen, K. J.; de Koster, R. (2001a):
Routing Methods for Warehouses with Multiple Cross Aisles. *International Journal of Production Research* **39**, 1865-1883.

Roodbergen, K. J.; de Koster, R. (2001b):
Routing Order Pickers in a Warehouse with a Middle Aisle. *European Journal of Operational Research* **133**, 32-43.

Rosenwein, M. B. (1994):
An Application of Cluster Analysis to the Problem of Locating Items within a Warehouse. *IIE Transactions* **26**, No. 1, 101-103.

Ruben, R. A.; Jacobs, F. R. (1999):
Batch Construction Heuristics and Storage Assignment Strategies for Walk/Ride and Pick Systems. *Management Science* **45**, 575-596.

Tompkins, J. A.; White, J. A.; Bozer, Y. A.; Frazelle, E. H.; Tanchoo, J. M. A.; Trevino, J. (1996):
Facilities Planning. 2nd ed., New York: John Wiley.

van Oudheusden, D. L.; Tzen, Y.-J.; Ko, H.-T. (1988):
Improving Storage and Order Picking in a Person-on-Board AS/R System: A Case Study. *Engineering Costs and Production Economics* **13**, 273-283.

van Laarhoven, P. J. M.; Aarts, E. J. H. (1992):
Simulated Annealing: Theory and Applications. Dordrecht et al.

Vaughan, T. S.; Petersen, C. G. (1999):
The Effect of Warehouse Cross Aisles on Order Picking Efficiency. *International Journal of Production Research* **37**, 881-897.

Wilson, H. G. (1977):
Order Quantity, Product Popularity, and the Location of Stock in a Warehouse. *AIIE Transactions* **9**, 230-237.