Improved storage in a book warehouse

Design of an efficient tool for slotting
the manual picking area at Wolters-Noordhoff

Master's thesis

Herwen de Ruijter

October 2007

Graduation committee:

Dr. P.C. Schuur
Ir. R.J. Mantel
Dr. Ir. H.B. Roelofs

Voorwoord

In het kader van mijn opleiding Industrial Engineering and Management aan de Universiteit Twente heb ik een afstudeeronderzoek uitgevoerd bij Wolters-Noordhoff in Groningen. Dit onderzoek was niet mogelijk geweest zonder de hulp en inzet van verscheidene personen. De docenten Peter Schuur en Ronald Mantel bedank ik voor hun uitstekende begeleiding. In tijden dat het onderzoek moeizaam verliep, hebben zij mij weer op weg geholpen. Tevens bedank ik Harold Roelofs, mijn begeleider vanuit Wolters-Noordhoff, voor zijn kritische blik op het onderzoek.
Tenslotte gaat een woord van dank uit naar mijn ouders, die mij in financiële en morele zin hebben gesteund tijdens het afstuderen.

Groningen, 2007

Herwen de Ruijter

Table of contents

Executive summary

This report is the result of a graduation project of the study Industrial Engineering and Management at Twente University. The research focuses on order picking at the warehouse of the educational publisher Wolters-Noordhoff. More than 350,000 orders (varying from very small to very large) are picked per year at the warehouse of Wolters-Noordhoff from three different areas: the bulk area, the dynamic picking area and the manual picking area. Order picking in the manual picking area is a costly process. Increasing the efficiency of this process can lead to considerable savings. Wolters-Noordhoff wants to increase the efficiency of the order picking process by (re)allocating products in the manual picking area in such a way that travel distance is minimized. This type of product allocation is called slotting. Therefore, goal of our research is to design a tool for efficiently slotting the manual picking area.

The research was executed in the following way. First, we analyzed the current situation and identified which factors affect order picking efficiency. We learned that warehouse employees either batch orders (i.e., combining several small orders into one batch and collect the batch in one picking tour) or split large orders into sub-orders. Furthermore, order pickers use a specific routing policy to collect the requested items. Momentarily, no specific slotting strategy is applied.

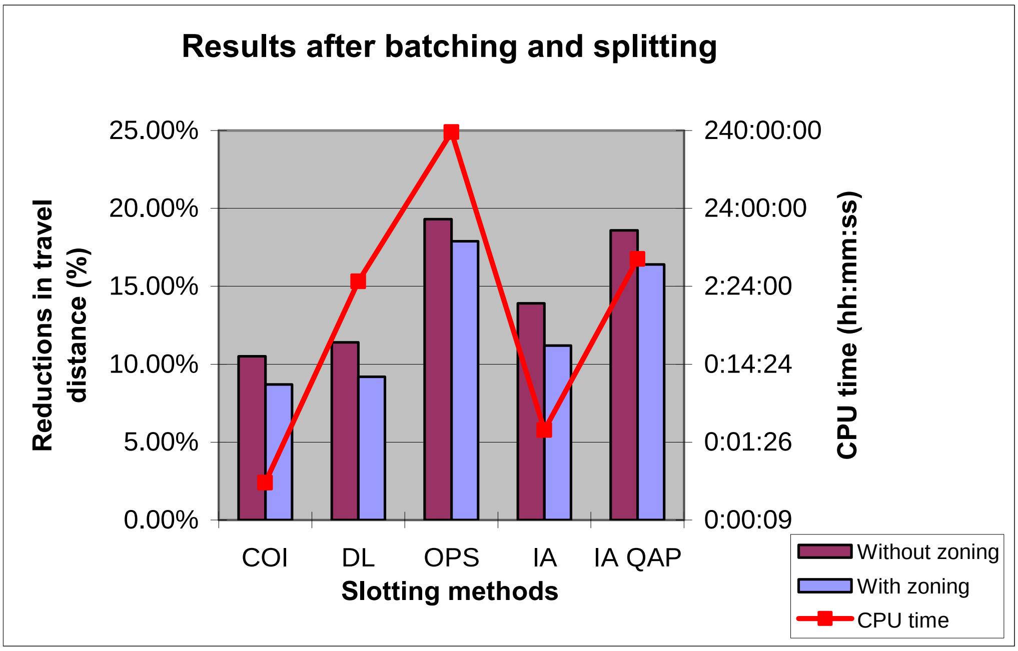
Next, we performed literature research in order to see what slotting methods can be used for Wolters-Noordhoff's situation. Many papers in the field of warehousing focus on operational decisions like batching and routing. Indeed, combining several orders into one batch and collect them in one picking tour can lead to an increased efficiency of the order picking process. In addition of course, shorter picking routes lead to time savings as well. Literature on slotting is rather scarce. A well known slotting strategy is to place frequently ordered products close to the depot (i.e., COI). This method is optimal in case order pickers collect one product per picking tour. This is not true in case multiple products are collected in one picking tour (as is the case with Wolters-Noordhoff). In those cases, it is more efficient to place products that are frequently ordered together close to each other. Cluster-based slotting methods identify 'closeness' relationships between (groups of) products. A shortcoming of these methods is that they do not explicitly work with travel distance. Instead, they use one of several surrogate measures of cluster strength. Furthermore, these researchers neglect the fact that there is a strong interrelationship between slotting and routing policies.

A first attempt to simultaneously consider these two issues is based on the idea of identifying how often two products are picked before or after each other (Direct Link method). Recently introduced slotting methods are based on the concept of order oriented slotting (OOS). OOS is based on multi-item orders and the accompanying methods directly relate the way of allocating items to locations to the chosen routing strategy for picking the orders. The interaction frequency heuristic (IA) ranks interaction frequencies of product pairs (i.e., the frequency that a product pair occurs on orders). Products with a high interaction frequency should be placed close to each other (relative to the routing-policy-specific distance), but also in accordance with their order frequency. However, it does not become clear when a product is placed in accordance with his order frequency. The interaction frequency quadratic assignment heuristic (IA QAP) tries to find a balance between placing product pairs with a high interaction frequency close to each other (relative to the routing-policy-specific distance) and placing frequently ordered products not too far from the depot. A parameter is used to find an appropriate balance between those two objectives. It is stated that this balance parameter must be determined empirically.

In the next phase of our research, we adapted the slotting methods extracted from literature (COI, Direct Link, IA and IA QAP) to the situation of Wolters-Noordhoff. We introduced a simple, but effective rule for the IA heuristic which clearly indicates when a product is placed in accordance with his order frequency. Furthermore, we introduced a rule of thumb for dynamically fine-tuning the balance parameter in the IA QAP heuristic.

We also developed a technique for solving the OOS problem: Order oriented product swapping (OPS). Slotting is a very difficult combinatorial optimization problem. We tackled this problem by using simulated annealing in combination with a clever neighbourhood structure. As opposed to other slotting methods, OPS directly minimizes travel distance.

It should be noted that slotting is strongly interconnected with the order batching and splitting problem. However, simultaneously considering batching, splitting and slotting is not very realistic and therefore we applied batching and splitting of orders *after* the slotting process (i.e., we first assume that every order is processed separately). Order pickers at Wolters-Noordhoff use their own experience to batch and split orders. Only a few 'hard' restrictions must be taken into account. We designed rules in such a way that the batching and splitting process is reflected as well as possible. It should be noted that batching diminishes the impact of slotting. This is not necessarily true for splitting.
Management prefers to place products belonging to the same product group close to each other (i.e., zoning). A reason amongst others is that order pickers are spread out more evenly over the manual picking area. Zoning can diminish the effect of slotting. The slotting tool also takes zoning into account.
In the last phase of this research, we constructed the slotting tool. We incorporated the different slotting methods in our tool. A sufficiently large order profile (i.e., 5 months of order data) is used to determine the efficiency of the slotting methods. Results (after batching and splitting) are shown below.



Results of all slotting methods after the batching and splitting process.
(DL = direct link method, OPS = order oriented product swapping, IA = interaction frequency heuristic, IA QAP = interaction frequency based quadratic assignment heuristic)

The figure above shows that even the most simple slotting strategy (i.e., COI) obtains a significant reduction in travel distance. In addition, we see that zoning does not have a huge impact on the amount of travel reduction. This is possibly due to the order structure. The direct link method performs rather disappointingly. The fact that the warehouse of Wolters-Noordhoff consists of multiple aisles possibly hurts the efficiency of the heuristic (the direct link method was originally applied to a single aisle). The order oriented slotting strategies (OPS, IA and IA QAP) perform very well. The current travel distance can be reduced by more than 15% both in cases with and without zoning applied. Our slotting technique (OPS) obtains the highest reduction in travel distance but at the cost of a large amount of computation time. The other order oriented slotting methods (IA and IA QAP) are computationally less intensive. They almost obtain the same reduction in travel distance. In addition, their computation time does not depend for a great deal on the size of the order profile as opposed to our own slotting heuristic (OPS).
Obviously, the current assignment of products to locations in the manual picking area is not very efficient. Since the order pattern is rather stable, we advice to implement the interaction frequency

based quadratic assignment slotting method. This method performs almost as well as OPS, but uses far less CPU time. Because the negative impact of zoning on the amount of distance reduction is minimal, we have no objections to the application of zoning.

Our slotting tool indicates the new ‘optimal’ locations of products. In addition, reallocation assignments of products are created. By incorporating the reallocation assignments generated by our slotting tool in the Warehouse Management System, order pickers can gradually move products to their new locations in quiet periods (e.g. the fall season).

The warehouse manager should use the slotting tool once or twice in order to determine if the profits (in terms of distance reduction) of again reallocating products are high enough.

1 Introduction

This thesis is the result of a graduation project of the study Industrial Engineering and Management at Twente University. The research focuses on the warehouse of the company Wolters-Noordhoff. This chapter gives a short description of the organization. An overview of the warehouse is also provided. In the next chapter, we describe the problem statement, the research method applied and give an overview of the outline of this thesis.

1.1. Organization

Wolters-Noordhoff publishes educational products for virtually every kind of educational institution. It was a subsidiary of Wolters Kluwer until 2006. In 2007, Wolters Kluwer sold its division to Bridgepoint Capital. Within the Dutch market, Wolters-Noordhoff is market leader and specialist in transforming information to educational material. This material covers printed as well as digital documents. Typical customer segments of Wolters-Noordhoff are: book stores, educational book stores, school suppliers, educational institutions, practical training organizations and individuals. Sales over 2006 amount to 124 million euro. The mission statement of Wolters-Noordhoff is as follows: *Learning for life with Wolters-Noordhoff.*
The market segments in which the organization is active, are: primary education, high schools, universities and professional education. Furthermore, Wolters-Noordhoff supplies atlases and additional educational products (e.g. educational games).
For every market segment, a so-called publishing group is responsible. Publishing groups develop new products for their market segments and are assisted by commercial divisions.
Wolters-Noordhoff has over 425 employees, who are divided over two locations in Groningen and one location in Houten. The warehouse is situated in Groningen and is described in the next section. An organizational chart of the main divisions is displayed in figure 1.

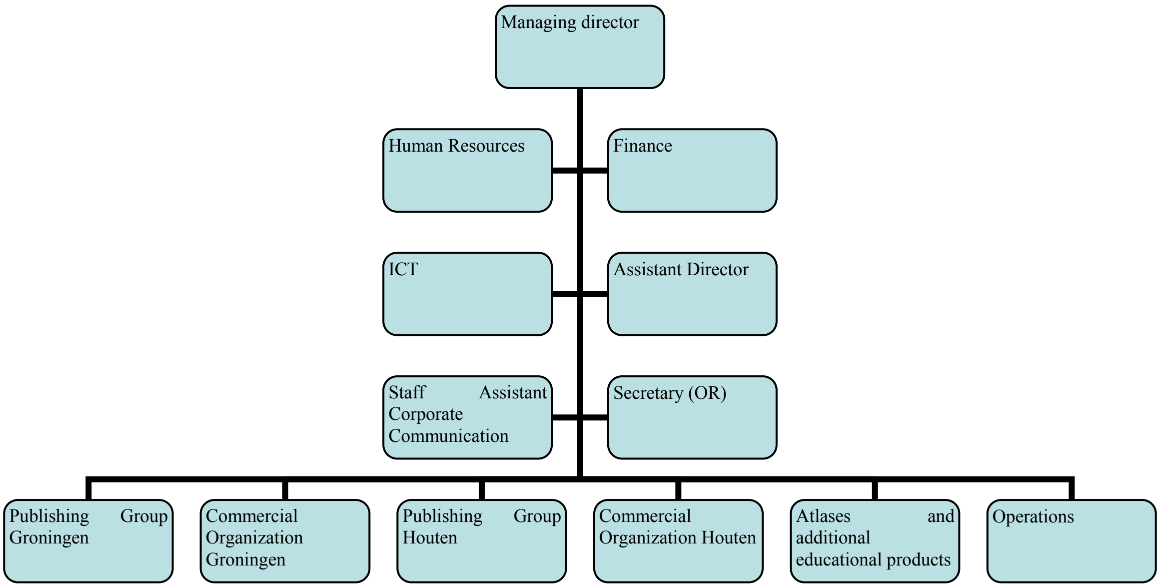


Figure 1: Organizational chart of Wolters-Noordhoff.

1.2. Warehouse

Wolters-Noordhoff has its own warehouse from where customer orders are handled. In addition, Wolters-Noordhoff uses Centraal Boekhuis (Culemborg) in order to supply small bookstores.
Product sales are strongly seasonal with a clear peak in the months May, June and July. During this peak, often more than 90,000 items leave the warehouse every day. The months August and

September are characterized by many supplementary orders with a large amount of order lines and small order quantities.
Warehouse personnel consist of 22 employees (excluding sales administration). Because of the seasonal sales, up to 35 temporary workers are hired during the peak months. These workers assist mainly in order picking activities. The warehouse is part of the operations division. Figure 2 displays the organizational chart of this division.

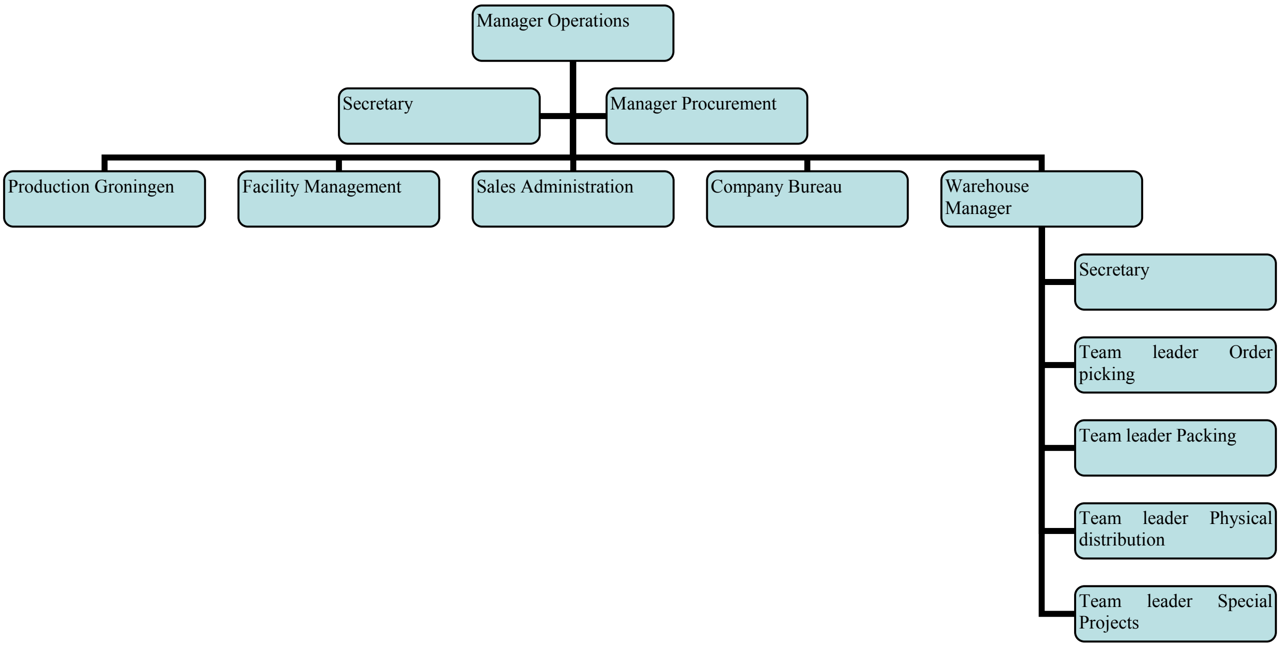


Figure 2: Organizational chart operations division.

The order picking activities performed at the warehouse are the field of research in this thesis. The next chapter provides the research objective as well as the research method and the outline of the thesis.

2 Research design

Order picking at Wolters-Noordhoff forms the field of research. A short description is provided in section 2.1. Next, the research objective is stated in section 2.2. followed by the research method in section 2.3. Finally, the outline of the thesis is provided in section 2.4.

2.1. Order picking at Wolters-Noordhoff

Order picking involves the process of retrieving products from storage (or buffer areas) in response to a specific customer request (De Koster *et al*., 2006). It is part of the logistics chain and has an important impact on the chain performance.

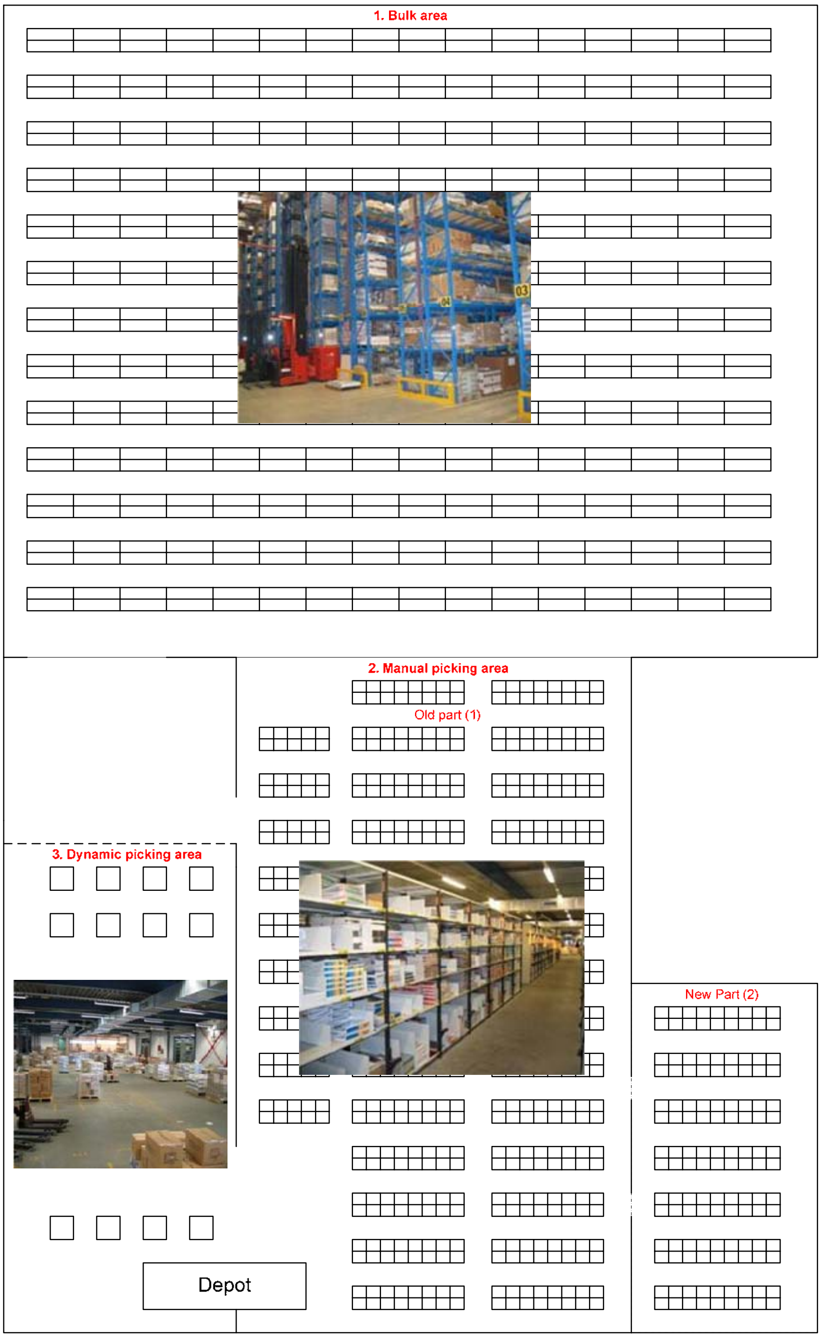


Figure 3: Overview of the picking areas.

At the warehouse of Wolters-Noordhoff, approximately 350,000 orders are picked per year. The number of order lines amounts to 1,300,000 per year. Orders consist of 1 up to several hundreds of order lines. About 9,000,000 items leave the warehouse every year. Orders are picked from three different areas:

1. bulk area (pallet picking)
2. manual picking area (manual picking)
3. dynamic picking area (manual picking)

An overview of these areas is provided in figure 3. The research focuses on the manual picking area.

2.2. Research objective

In order to reduce supply chain costs, Wolters-Noordhoff wants to improve order picking efficiency. Order picking time must be reduced with the ultimate goal of reducing the amount of temporary workers. Several warehousing decisions like the sequencing of picks per order and batching methods influence order picking efficiency (Mantel and Rouwenhorst, 1998). Wolters-Noordhoff is interested in the application of an optimized storage assignment strategy (i.e., slotting) of products for the manual picking area. In manual order picking systems, travel time is an increasing function of the travel distance (Hall, 1993). Therefore, the objective of this research is as follows:

Determine a slotting strategy for the manual picking area with the goal to minimize travel distance.

The objects of research are order picking at the manual picking area and application of slotting strategies. The research model is depicted in figure 4.

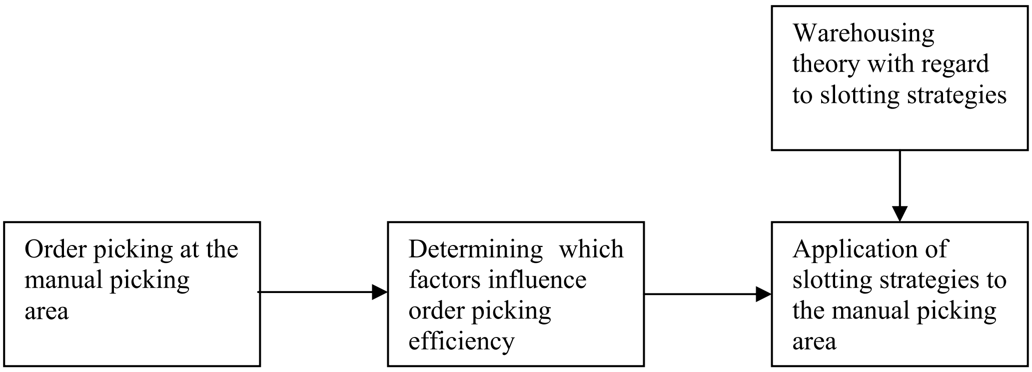


Figure 4: Research model.

As follows from this model, three research questions must be answered:

1. Which activities have a large impact on order picking efficiency?
2. Which slotting strategies can be applied to the current situation and which one is the most efficient?
3. How can the proposed slotting strategy be implemented?

The first question focuses on the current order picking process. Activities that have a large effect on order picking efficiency need to be addressed adequately.
To answer the second question, warehousing literature with regard to slotting needs to be investigated. If no suitable slotting strategy exists, a new one needs to be developed. An efficient slotting strategy obtains a significant reduction in travel distance within a reasonable amount of computation time. In order to compare the current and new situation, relevant order data and product data must be gathered. Finally, issues concerning the implementation of the proposed slotting strategy need to be handled adequately.

2.3. Research method

This research is executed in the following way. We start with an analysis of the order picking process by interviewing warehouse employees. Documented working methods are analyzed as well. As a result, we can identify which elements have a significant impact on the traversed distance in the manual picking area. After these steps, we perform literature research concerning slotting. We select viable slotting strategies from literature and design a new one. Physical restrictions of the manual picking area like the lay-out as well as certain wishes from management concerning the allocation of products are taken into account. We collect order data that represents the order pattern. After these steps, we are able to determine the current traversed distance given the provided order data. We incorporate the previously selected slotting strategies in our tool and analyze the results. Based on these results, we select the most efficient slotting strategy for the warehouse of Wolters-Noordhoff. The warehouse manager can use this slotting tool. He can insert order data, alter various assumptions we made and analyze the results. The slotting tool also indicates the new (optimized) locations of products.

2.4. Outline of the thesis

In chapter 3, order picking at Wolters-Noordhoff is described in detail. An overview of warehousing literature with respect to slotting strategies is given in chapter 4. Strengths and weaknesses of previously conducted research are also discussed. In chapter 5, we select slotting strategies and adapt them to the current situation. Various restrictions and assumptions we made concerning slotting at the manual picking area are described. We incorporate existing slotting methods from literature as well as a new slotting strategy for the manual picking area in our slotting tool and provide the results in chapter 6. Based on a comparison of these results, we select the most efficient slotting strategy. Conclusions and recommendations can be found in chapter 7. Implications of this research and suggestions for further research are also given in this chapter. Finally, in chapter 8 we provide a reflection on this project.

3 Current situation

A detailed overview of order picking and related logistic activities at Wolters-Noordhoff is given in this chapter. Other logistic activities at the warehouse are beyond the scope of this research and thus are not discussed.

3.1. Picking areas

Orders are picked from three different areas: 1. bulk area, 2 manual picking area. and 3. dynamic picking area (figure 3). In the bulk area, products are stored on pallets in high selective pallet racks. Man-on-board AS/RS equipment is used to store and retrieve pallets. In order to reduce space, narrow aisles are used. There are approximately 15,000 pallet locations.
The manual picking area consists of an old (1) and a new part (2). Addition of the new part took place in 2003. Throughout the remainder of this thesis, we denote the old part of the manual picking area as *part 1* and the new part as *part 2*. In appendix I, these numbers are shown as well. About 850 shelves are located in part 1 and 160 shelves are located in the part 2. Every shelf consists of a certain amount of compartments. One product can be stored in each compartment in relatively small amounts (smaller than pallet amounts). There are about 12,000 compartments. Compartments are classified by volume (section 3.3). Items are manually retrieved with pick carts. Shelves are equipped in such a way, that the largest compartments are situated at or below waist height in order to facilitate order picking. Temporary locations are created in the dynamic picking area in order to reduce travel time. In section 3.2., we elaborate on this issue. Pallets are retrieved from the bulk area and stored at the dynamic picking area. At the end of a working day, the pallets are put back in the bulk area. Order picking at the dynamic pick area is performed manually with the aid of pick carts.

3.2. Order picking process

Figure 5 depicts the main process steps. Several of these steps are performed with the aid of a Warehouse Management System (WMS). A WMS is software designed specifically for managing the movement and storage of materials throughout the warehouse. The main process steps will now be explained. First, the WMS checks the amount of inventory of all products that need to be picked on the current day. If the inventory for a certain product is insufficient, the accompanying order lines of all orders for that specific day are left out on the pick lists (naturally, those order lines are not deleted from the WMS and will be released in the near future when the amount of inventory for the accompanying products *is* sufficient). After the inventory check, the orders are released for order picking. Dynamic locations are created based on the item quantities demanded. Furthermore, replenishment tasks are generated for the manual picking area. We illustrate these two steps with an example. Consider the following set of orders, that need to be picked:

Order 1		Order 2		Order 3		Order 4	
product	quantity	product	quantity	product	quantity	product	quantity
X	10	X	20	Z	120	X	10
Y	20	Y	30			Y	40
Z	10						

Inventory data is stated below.

product	pallet size bulk	capacity manual picking area *	current inventory manual picking area
X	50	10	5
Y	60	20	10
Z	100	40	2

(*Because of possible return of goods, the total capacity for a certain product in the manual picking area will be slightly higher.)

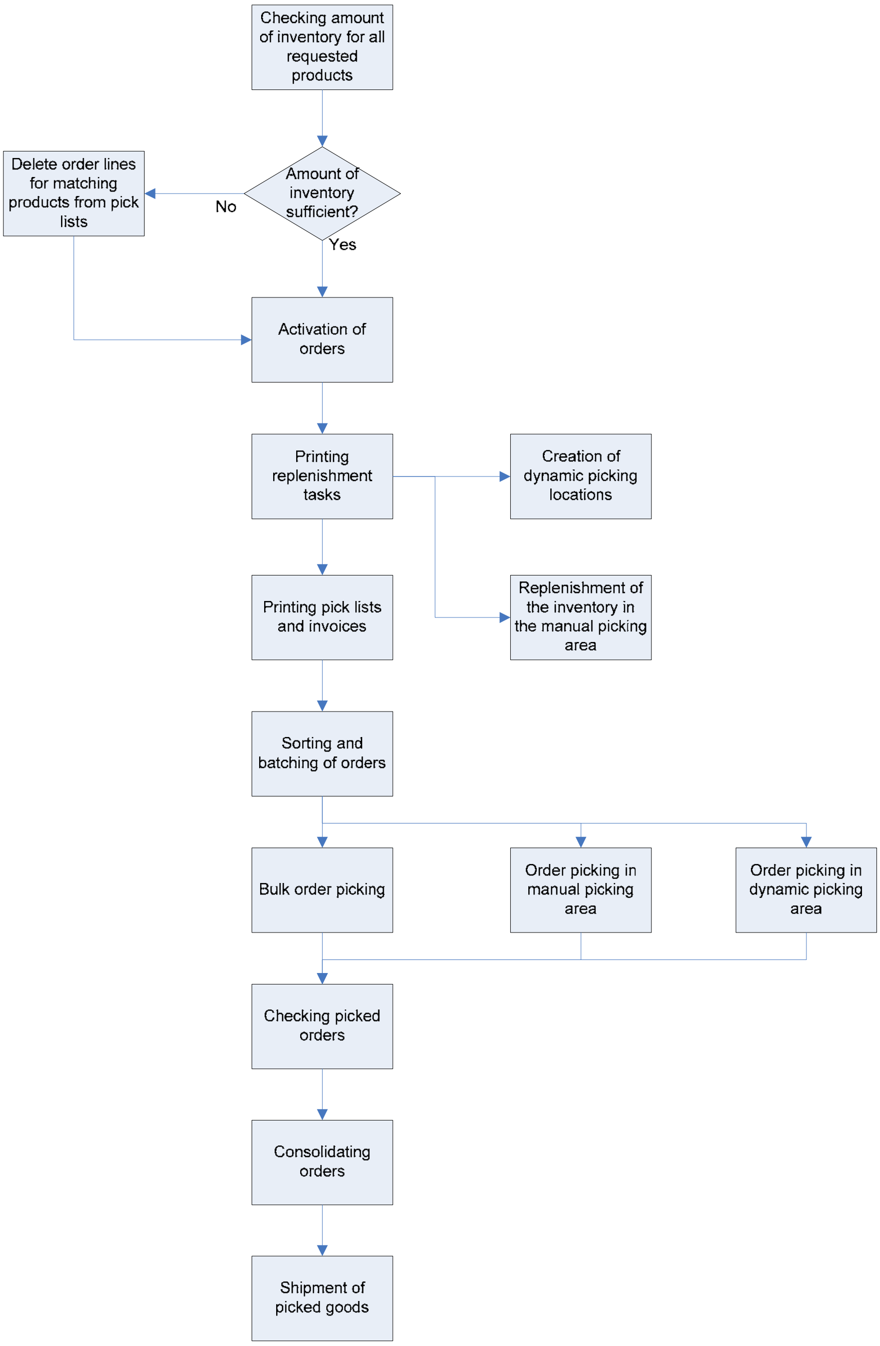


Figure 5: Flowchart of the order picking process.

The total demand for item X amounts to 40 pieces. This exceeds the capacity for the manual picking area (10 pieces). Thus, the WMS creates a dynamic location. A warehouse employee moves one pallet with item X (50 pieces) to the dynamic picking area. The same line of reasoning can be applied to item Y. Total demand is 90 pieces and this exceeds the capacity in the manual picking area (20 pieces). The WMS generates a dynamic location and the goods are moved to the dynamic picking area. Note that in this case two pallets are needed.
The order quantity for item Z amounts to 130 pieces and again this exceeds the capacity of the manual picking area. However, no dynamic location is generated because of the following reason. The order quantity of order 3 is 120 pieces and this is more than the pallet size of item Z (100 pieces). Thus, one pallet with item Z will be picked straight from the bulk area. The amount of remaining items (20 for

order 3 and 10 for order 1) does not exceed the capacity of the manual picking area (40 pieces). However, current inventory of item Z is too low and thus item Z must be replenished by an amount of 40 – 2 = 38 pieces. An order picker will replenish item Z before he / she starts with order picking.
Concluding, a dynamic location is created when the total demand of a product exceeds the capacity in the manual picking area for that product. However, if the demanded item quantity of one order exceeds the pallet size, the items will be picked directly from the bulk area. Creation of dynamic locations and replenishment of inventory are performed prior to order picking. The goal of using dynamic pick locations is to reduce order picking time. Indeed, frequently demanded items are placed in a relatively small area close to the depot in order to reduce travel time.
In the next step, pick lists and invoices are printed. There exist two types of pick lists:

1. Manual pick lists (dynamic and manual pick locations)
2. Bulk pick lists (bulk locations only)

Every pick list consists of a number of order lines. An order line states the product code, location code and the demanded item quantity.
Order lines on a pick list are sorted on location codes per pick area (see section 3.3 for an explanation of location codes). Note that bulk picking is performed separately, because item quantities are in pallet sizes and specific equipment must be used to retrieve pallets.
After the pick lists and invoices have been printed, order pickers batch and split the orders. Order batching is the method of grouping a set of orders into a number of sub-sets, each of which can then be retrieved by a single picking tour (De Koster *et al.*, 1999). However, there is also a small percentage of orders with a large amount of order lines. These orders are split up in smaller parts. Each part can then be retrieved by a single picking tour. Order pickers batch and split orders based on certain restrictions and their own experience. The basic restrictions are as follows:

1. An order with a weight of more than 10 kg can not be added to a batch which has an order that has a weight of less than 10 kg.
2. Only orders with the same dispatch method (TNT, Centraal Boekhuis, internal) may be added to the same batch.

Except for the restrictions mentioned above, order pickers are free in choosing batching and splitting methods. We will come back to this issue in chapter 5.

3.3. Order picking in the manual picking area

The following step consists of manually scanning the paper pick lists with a barcode scanner which communicates with the WMS. The WMS sorts the accompanying order lines on increasing location code. In appendix I, a detailed map of the manual picking area can be found. A location code is composed of six digits. The first two digits indicate the aisle number. The next two digits represent the shelf number and the last two digits indicate the compartment number in the shelf (level and sub-location). Figure 6 gives a clear example of a location code in the manual picking area. The scanner determines the initial pick sequence. However, the order picker has the possibility to change this pick sequence by reshuffling the location codes. This way, shorter or more logical routes can be obtained (at least observed by the order picker). The specific routing policies are discussed in chapter 5.
In the next step, an order picker chooses an appropriate pick cart at the depot and starts a picking tour. As stated in section 3.1, pick carts are used for order picking in the manual and dynamic picking area. Two types of pick carts are used: small carts (for orders with a weight of less than 10 kg) and big carts (for orders with a weight of more than 10 kg). An order picker sorts the items by order while conducting the picking tour. This is known as the 'sort-while-pick' principle (De Koster *et al.*, 1999). The order picker starts at the manual picking area, collects the requested items and then collects the items located at the dynamic picking area. The dynamic picking area is near the depot (figure 3).
Depending on the season, all orders or a sample of orders are checked. In this phase, split orders are also consolidated. After inspection, the goods are moved to the packing area.

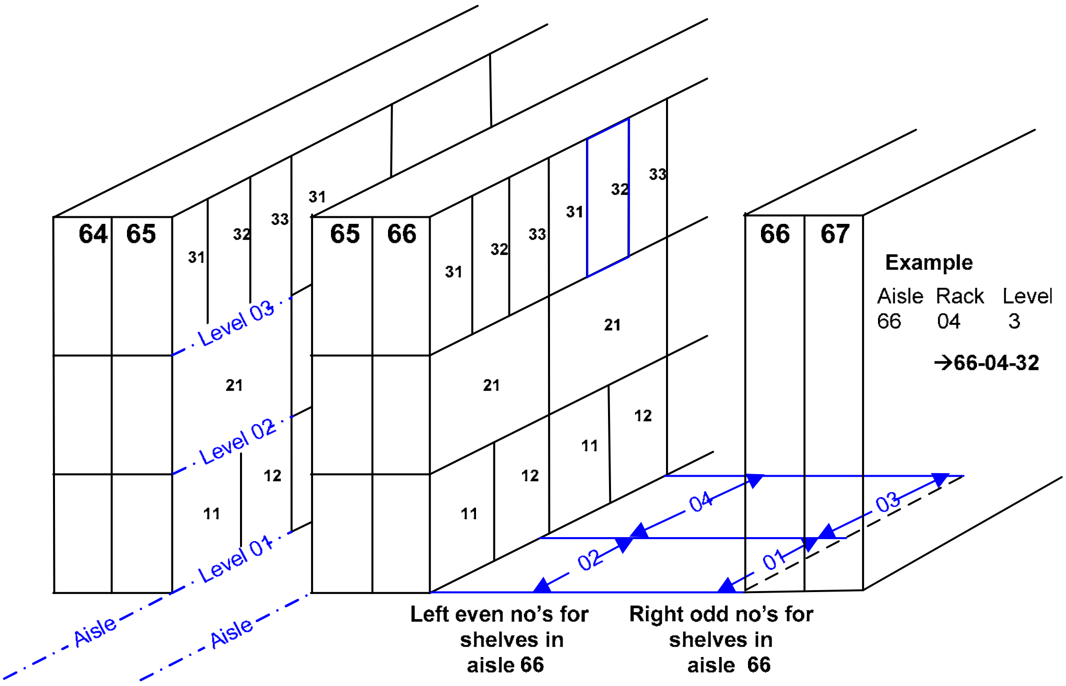


Figure 6: Example of location codes in the manual picking area. Note that the largest compartments (in terms of volume) are situated at or below waist height in order to facilitate order picking.

3.4. Current slotting strategy

The slotting strategy describes the activities associated with optimizing product placement in locations in a warehouse. Another commonly used name for this activity is storage assignment policy. At Wolters-Noordhoff, there are currently no specific policies for the allocation of products in the manual picking area. Team leaders (cf. the organizational chart) are responsible for product allocation. Based on size of the products and expected turnover, a specific compartment is chosen (cf. appendix II for an overview of compartment sizes). Each product belongs to a specific segment:

1. Elementary school
2. High school
3. University
4. Professional training
5. Additional educational products

In previous years, products were allocated based on their segment. Currently, only elementary school products are placed close to each other. Aisles 84 up to 93 are used for these products (cf. appendix I). The remaining segments (high school, university, professional training and additional educational products) are placed in aisles 1 up to 81. No specific historical order data (e.g., order frequencies) is used for storage assignment.

4 Literature research

Bartholdi and Hackman (2006) point out that approximately 65% of warehousing costs are related to order picking. Order picking can either be done automatically or manually. Roughly 80% of order picking systems in Western Europe consist of so-called manual picker-to-part systems. This means that an order picker walks or drives along aisles in a warehouse to pick items. Figure 7 shows typical order pick activities and the distribution of an order picker's time (Tompkins *et al*., 2003).

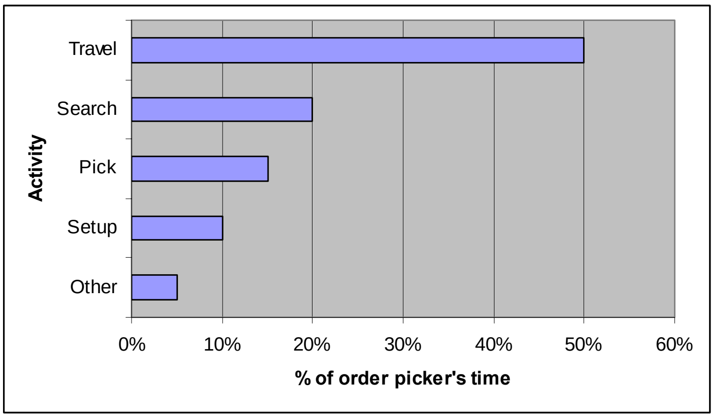


Figure 7: Distribution of an order picker's time (Tompkins *et al.*, 2003).

Traveling is considered as 'waste' (Bartholdi and Hackman, 2006), because it costs labour hours and does not add value to the product. Because it consumes a large part of an order picker's time, it is often a first candidate for improvement. Other important objectives include the minimization of total cost (both investment and operational) and the minimization of the throughput time of an order. Decisions made on design and control of order picking systems (both tactical and operational) have a major impact on these objectives. Common warehousing decisions at tactical and operational levels are (Mantel and Rouwenhorst, 1998):

1. Lay-out design and dimensioning of the storage system.
2. Assigning products to storage locations (slotting / storage assignment).
3. Assigning orders to pick batches and grouping aisles into work zones (batching and zoning).
4. Order picking routing.
5. Accumulation/sorting of orders.

Order picking has received a considerable amount of attention from researchers. Many advanced techniques and solution methods have been proposed for the warehousing decision problems mentioned above. There is a growing interest from warehousing companies in applying these methods. As we already stated, Wolters-Noordhoff wants to reduce travel distance in the manual picking area by reallocating products. Therefore, we focus on slotting strategies. We first give a short introduction in section 4.1 on storage assignment policies in general. Because slotting and routing are related, section 4.2 describes commonly used routing policies in warehouses. Order batching influences the effect of slotting. Therefore, a small introduction to batching is given in section 4.3. Next, several articles with respect to slotting are discussed. We conclude this chapter with some conclusions and indicate which slotting strategies are viable for Wolters-Noordhoff.

4.1. Introduction

Commonly used storage policies are either dedicated or random. A dedicated storage policy prescribes a particular location for each product, whereas with random storage a location is randomly chosen. In practice, if warehouse employees are free in choosing a location for a new product type, they will choose a location close to the depot. As a result, the occupancy of aisles further from the depot will decrease (i.e., more free locations). This method is also referred to as closest-open location. Thus, true random storage can only be obtained with a computer system (e.g. a Warehouse Management System) (De Koster *et al.*, 2006). Randomized storage needs less storage space than dedicated storage. Dedicated storage is applied in most warehousing applications. In between dedicated and random storage, a class based storage policy allocates zones to specific product groups, often based upon their turnover rate (Mantel and Rouwenhorst, 1998). Products are then allocated randomly within one zone. This way, one tries to incorporate the advantages of both storage policies, small travel distances and reduction of storage space.

Other more advanced storage policies are based on family grouping. These methods aim at storing products close to each other if they are often jointly requested by customers. In comparison to other topics on warehousing (e.g. routing and batching), the amount of literature on storage assignment policies is modest. However, researchers do acknowledge the importance of a clever slotting strategy (Petersen and Aase, 2003).

4.2. Routing policies

The objective of routing policies is to sequence the items on the pick list to ensure a short route through the warehouse. Ratliff and Rosenthal (1983) provide an algorithm for creating optimal (i.e., minimal distance) routes in a warehouse consisting of parallel aisles, one front cross aisle and one back cross aisle (figure 8). They show that they can solve the problem in running time linear in the number of aisles and the number of pick locations. In practice, often heuristics are used for routing order pickers. Many warehouses have rather specific lay-outs. As a result, the algorithm of Ratliff and Rosenthal can not be used for those situations. Furthermore, an optimal route may be perceived as illogical to an order picker. He or she creates a deviating route that is easier to follow. Commonly used routing policies are the return, S-shape, midpoint and largest gap strategy. Examples are provided in figure 8 (De Koster *et al.*, 2006). For the return strategy, an order picker enters and leaves a picking aisle from the same end. Using a S-shape policy, an order picker traverses each aisle that contains at least one pick location entirely. Aisles that do no contain pick locations, are not entered. After visiting the last aisle, the order picker returns to the depot. This type of routing policy is often applied in practice. The midpoint heuristic splits the warehouse in half. Picks in the front half are accessed from the front cross aisle and picks in the back are entered from the back cross aisle. Only the first and last aisles are traversed entirely if there are items in the half opposite to the depot. The largest gap strategy resembles the midpoint strategy, but in this case an order picker enters an aisle as far as the largest gap within an aisle. The gap represents the separation between any two adjacent pick locations (in the same aisle), between the first pick and the front aisle or between the last pick and the back aisle. If the largest gap is between two adjacent picks, the order picker performs a return route from both ends of the aisle. Otherwise, a return route from either the front or back aisle is used. Thus, the largest gap is the fraction of the aisle that is not traversed. These methods were originally developed for single block warehouses (i.e., warehouses with only a front and back cross aisle and no additional cross aisles). Roodbergen *et al.* (1998) have adjusted these methods in such a way that they can be used for multiple-block warehouses as well. The effect of a certain slotting strategy depends on the routing policy used. This relationship is described in section 4.6 and 4.7.

4.3. Order batching

Large orders (i.e., in terms of volume or weight) are often picked on individual basis. This is called single order picking (SOP). In case orders are smaller than the capacity of picking equipment (e.g. a pick cart), travel time can be reduced by picking a set of orders in a single picking tour. This is called order batching or simply batching. For manual picking systems, literature distinguishes basically two types of order-batching heuristics (De Koster *et al.*, 1999): seed and savings algorithms. Seed algorithms construct batches in two phases: seed selection and order congruency. A seed selection rule defines a seed order for each batch. For example, a seed order can be a random order, an order with a large amount of order lines or an order with the longest picking tour. An order congruency rule

determines which unassigned order should be added to the current batch. Examples are (De Koster *et al.*, 1999): the number of additional aisles that have to be visited if the order is added or the sum of the travel distances between every location of an item in the order and the closest location of an item in the seed order. Savings algorithms are based on the algorithm of Clarke and Wright (1964) for the vehicle routing problem. A saving on travel distance is obtained by combining a set of small tours into a smaller set of larger tours. De Koster *et al.* (1999) compares seed and time savings heuristics for commonly used routing policies. They conclude that even the most simple batching methods significantly reduce travel time compared to single order picking.
Batching influences the effect of slotting. If picking equipment has a large capacity and many orders can be inserted into a relatively small amount of batches, slotting does not achieve huge reductions in travel distance / time. Indeed, collecting a batch of orders means that a large amount of picking locations have to be visited. As a result, a clever allocation of products is of little importance.

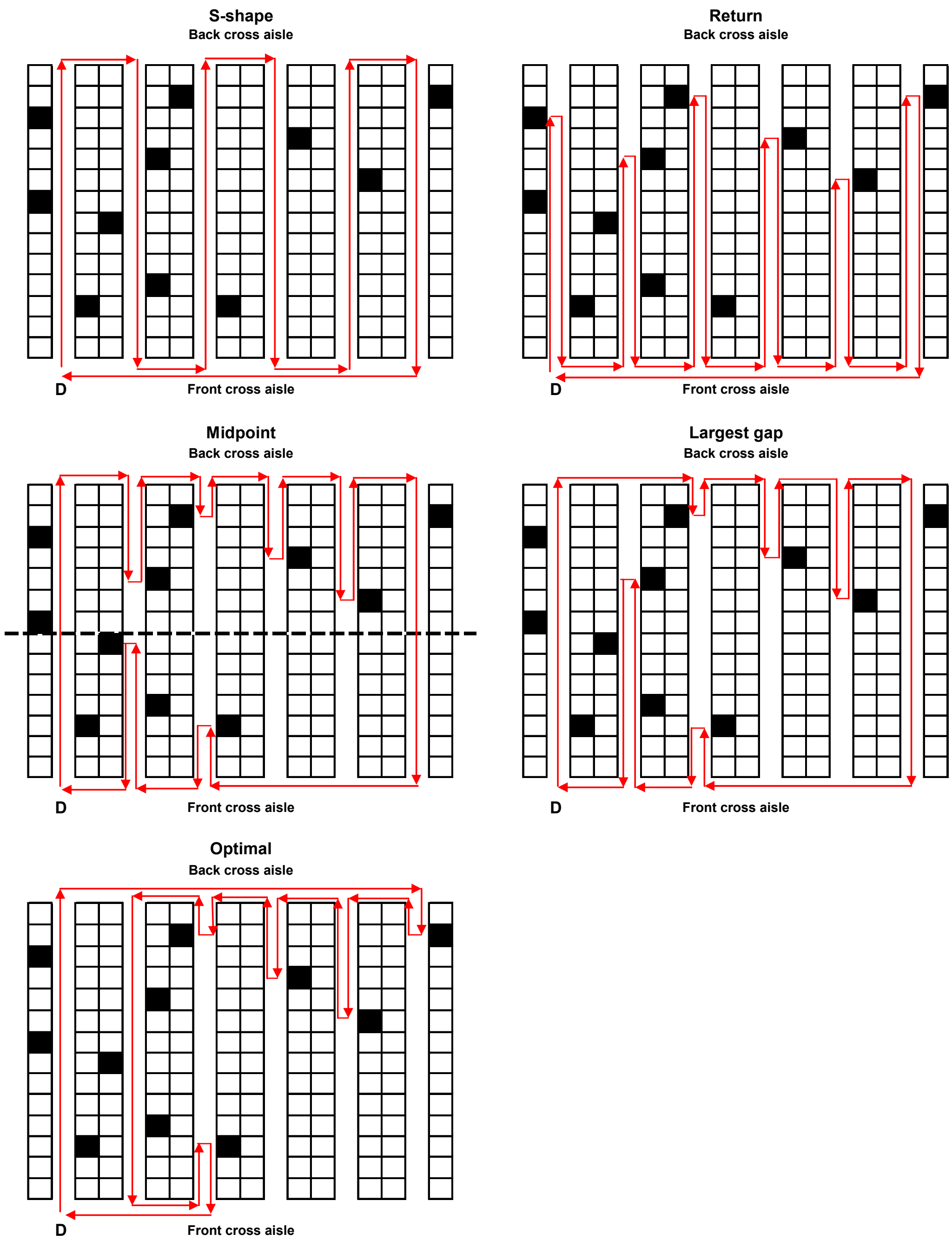


Figure 8: Examples of routing policies. Black squares indicate pick locations. The depot is located bottom left. (De Koster *et al.*, 2006).

4.4. Cube per order index (COI)

Order pickers can either pick one product type per route (single command) or multiple product types (dual / multi-command cycles) In practice, single command cycles occur at warehouses where retrieval equipment only has storage space for 1 product type (e.g., pallet trucks or unit-load AS/RS-equipment). The slotting problem in this case can be formulated as an integer linear programming model (Heskett, 1963). Let us first introduce the following variables:

q = number of storage locations
n = number of products
m = number of input/output (I/O) points
S_j = number of storage locations required for product j
T_j = number of trips in/out of storage for product j (e.g., throughput of product j)
p_i = percentage of travel in/out of storage to/from I/O point i
d_{ik} = distance required to travel from I/O point i to storage location k
x_{jk} = 1 if product j is assigned to storage location k, 0 otherwise
$f(x)$ = average distance traveled

The integer linear programming model is as follows:

$$\min z = \sum_{j=1}^{n}\sum_{k=1}^{q}\frac{T_j}{S_j}\sum_{i=1}^{m} p_i d_{ik} x_{jk}$$

s.t.

$$\sum_{j=1}^{n} x_{jk} = 1 \qquad k = 1....q \quad (1)$$

$$\sum_{k=1}^{q} x_{jk} = S_j \qquad j = 1...n \quad (2)$$

$$x_{jk} = (0,1) \qquad \forall j,k \quad (3)$$

Restriction 1 ensures that exactly one product is assigned to each storage location. The second restriction ensures that the assigned storage locations for a product match the required number of storage locations. The problem presentation above corresponds to a form of the balanced transportation problem. Implicitly it has been assumed that: $q = \sum_j S_j$. For the problem to be feasible, in general, it must hold that: $q \geq \sum_j S_j$. If $q - \sum_j S_j > 0$, the previous balanced formulation is obtained by introducing a fictitious product 0, with $S_0 = q - \sum_j S_j$ and $T_0 = 0$ (Warehouses have often more storage locations than product types). The above transportation problem can be solved to optimality by using the following solution method.

1. Compute $f_k = \sum_{i=1}^{m} p_i d_{ik}$
2. Renumber locations by $f_1 \leq f_2 \leq ... \leq f_q$
3. Renumber products by $\frac{T_1}{S_1} \geq \frac{T_2}{S_2} \geq .. \geq \frac{T_n}{S_n}$
4. Assign locations 1,2,..,S_1 to product 1, locations S_1 + 1, S_1 + 2, .., S_1 + S_2 to product 2, etc

Heskett (1963) uses the solution method above by introducing the Cube-per-Order-Index (COI):

$$COI_j = \frac{S_j}{T_j}$$

The COI-index is calculated for each product and the corresponding values are sorted in increasing order. Next, minimal distances from each location to the depot are determined taking into account the specific aisle structure of the warehouse. Products with the lowest index are placed at those locations nearest the depot. In the majority of warehousing situations, multi-command cycles are executed. Multiple items are picked (or stored) in one picking tour. The COI-rule neglects the composition of an order. Thus, it is expected that only sub-optimal solutions can be found. However, for a rather specific scenario, Malmborg (1989) shows that application of the COI-rule leads to an optimal solution (in terms of minimal travel time) in a multi-command situation as well. His research addresses warehouses where aisle-captive AS/RS-equipment is used. It is noted that Malmborg assumes there is no demand correlation between products. In practice, this does not often occur.
Although the COI-rule guarantees no optimal solutions for multi-command order picking, many warehouses apply this storage strategy because of its practicability. Research (Petersen and Aase, 2003) shows that using the COI-rule in multi-command order picking systems still can lead to a (sometimes considerable) reduction in order picking time. The amount of reduction depends on the routing and batching strategy used. COI can also be applied without considering the volume of products.

4.5. Correlated storage methods

This class of methods tries to take into account the correlation between products, which is documented by the fact that certain pairs of product types are demanded together more frequently and therefore appear more often together in the same customer/picking order than others. These methods cluster product types into groups according to some measure of the strength of the joint demand. Items belonging to the same cluster are placed as 'close' together as possible. The slotting problem in those cases is divided in two sub problems:

1. Identification of clusters of item types (clustering problem).
2. Assignment of clusters to locations (location-assignment problem).

Frazelle (1989) is the first researcher who has tried to tackle this type of the slotting problem. He proposes a heuristic approach to group items into clusters. He uses a correlation measure stating the conditional probability that two items appear on one order. Independence hypothesis tests are used to filter out pairs with low correlation. He starts with the most popular product (e.g., product with the highest probability of appearing on an order) and adds this to an empty cluster. Of all other products that are correlated, the one with the highest total correlation is added to that cluster. Products are no longer added to the cluster when a certain capacity constraint is violated. After a predetermined number of clusters are generated, he places the clusters with highest total popularity closest to the depot (second sub problem). This last step resembles the COI-concept.
Rosenwein (1994) shows that the clustering problem can be formulated as a p-median clustering problem. The p-median cluster model is originally used as a facility location model with an additional constraint (p facilities to be opened). The number of facilities to be opened or clusters to be formed (p) in a p-median cluster problem is determined on a priori information (e.g. number of aisles). On the basis of a distance matrix **D**, different elements (medians) will be selected to which the rest of the elements will be allocated, so that p clusters are created. Afterwards the value of p may be changed, and the procedure may be repeated. A cluster median is defined as the element j that is representative for all elements in the cluster. Because p clusters must be created, there are also p medians, with $1 \leq p \leq m$ where m is the number of products which have to be clustered. The following variables and parameters are defined.

$$x_{ij} = \begin{cases} 1 \text{ if product } i \text{ is allocated to median } j \\ 0 \text{ otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 \text{ if product } j \text{ is a median} \\ 0 \text{ otherwise} \end{cases}$$

d_{ij} = distance, time or cost between products i and j
q = customer order
p = number of medians
I = set of products that have to be clustered
J = set of products that can act as medians
Q = set of customer orders

Rosenwein defines d_{ij} for the location assignment problem as follows: $\sum_{q \in Q} \left| v_{iq} - v_{jq} \right|$. The variable v_{iq} is binary and indicates whether product type i is included in order q:

$$v_{iq} = \begin{cases} 1 \quad \text{if product type } i \text{ is included in order } q, q \in Q \\ 0 \quad \text{otherwise} \end{cases}$$

Thus, a small value of d_{ij} indicates that product types i and j are frequently ordered together. Indeed, if product i and j always appear together on one order, d_{ij} is equal to 0. The p-median clustering model is now as follows.

$$\min z = \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \qquad (1)$$

s.t.

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \qquad (2)$$

$$\sum_{j \in J} y_j = p \qquad (3)$$

$$x_{ij} \le y_j \quad \forall i \in I, \forall j \in J \qquad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i \in I, \forall j \in J \qquad (5)$$

By means of the p-median problem all product types are assigned to p clusters and from each cluster one product type is selected as a median, such that the total *formulated* distance (d_{ij}) between the median and accompanying product types is minimized (c.f. objective function (1)). Constraint (2) ensures that all elements are allocated to medians /clusters. Restriction (3) ensures that exactly p medians / clusters are formed. Constraint (4) links elements and medians. Constraint (5) ensures that the variables x_{ij} and y_{ij} are binary. Rosenwein also introduces an approximation of the total *travel* distance, which is dependent on the number of clusters. Hence, the aim is to minimize the number of clusters. The subsequent sub problem, allocating clustered items to locations, is not discussed in the article.
Amirhosseini and Sharp (1996) introduce an *order-satisfying correlation measure (OSCM)* in order to solve the clustering problem. This measure looks at the degree to which two or more products together fill warehouse or customer orders. They also propose a clustering method that merges the attributes (outcomes of the OSCM) from the cluster with the new product added to it so at each stage the original cluster is nested inside the new one. The motivation for introducing this clustering method is to minimize travel distance. However, the authors do not indicate how the product clusters should be allocated in a warehouse and it is unclear what amount of travel reduction can be obtained by using this method.

Liu (1999) proposes a similar solution method as Rosenwein (1994) for the clustering problem, but he uses a different formula for the similarity of products *i* and *j* ('distance' between product *i* and *j*):

$$c_{ij} = \frac{1}{|Q|}\sum_{q \in Q} \frac{\min[u_q^i, u_q^j]}{\max[u_q^i, u_q^j]}$$

Again, Q denotes the set of all customer orders and q represents a customer order. In this case, d_{ij} denotes the similarity between products i and j, u_q^i is the demand (i.e., amount of items) required of product i in customer order q. The variable c_{ij} can be interpreted as the probability that i and j appear together in the same order. Naturally, if $c_{ij} = 0$, product types i and j never appear on the same order. Hence, Liu tries to *maximize* the sum of formulated distances for the p-median problem. As opposed to Rosenwein (1994), Liu gives no indication on how to determine the number of clusters. In order to solve the second sub problem, Liu proposes to identify the product type with the largest order quantity and assign it to a location closest to the depot. All other product types that are located in the same cluster are assigned to locations according to the standard frequency based location strategy (COI-based). These steps are repeated until all product types have been assigned. Liu provides an example with only one picking aisle. However, for a warehouse with multiple picking aisles, it is likely that clusters must be divided (especially with a small number of large clusters). This problem also occurs, if product specifications (e.g. volume) vary and products must be placed in specific locations. The division of clusters contradicts the original idea of a clustering method.
Garfinkel (2005) explores the first sub problem (clustering) in detail but uses a different approach. He starts from a predetermined number of zones. In each zone, a certain amount of product types can be stored. His research concentrates on the allocation of product types to zones in such a way that the amount of entered zones for a given set of orders is minimized. This objective deviates from the common objective of slotting. An integer linear programming model is formulated for the general problem. As opposed to many other authors, Garfinkel also considers the cost of reallocating product types between zones (re-warehousing costs). Because of NP-completeness, the problem cannot be solved to optimality for large data instances. Therefore, heuristics based on graph partitioning theory are used. Obviously, an important question remains how product types should be allocated per zone. Garfinkel assumes that every zone must be traversed completely. Thus, the second sub problem is not considered.

4.6. Slotting based on storage allocation patterns

Instead of considering the exact distances between the available locations and the depot as with the COI-rule, several researchers use 'storage allocation patterns'. According to these patterns, items are assigned to locations. Commonly used allocation patterns are diagonal storage, within-aisle storage and across-aisle storage. These storage allocation patterns are depicted in figure 9.

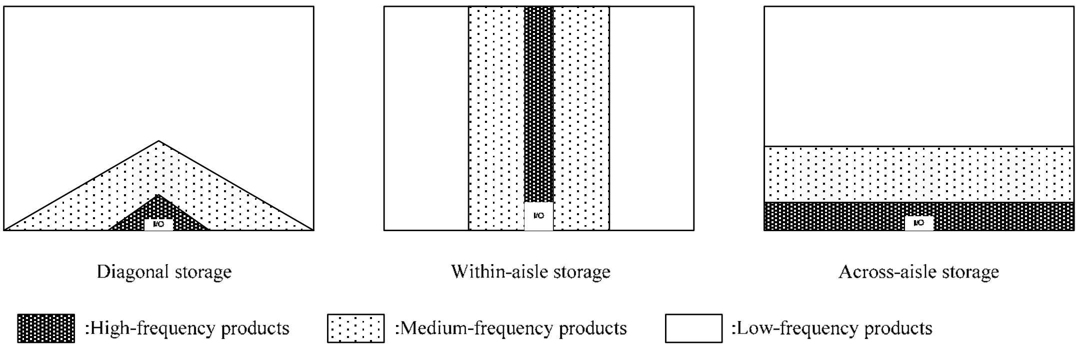


Figure 9: Storage allocation patterns.

They indicate where high-frequency, medium-frequency and low-frequency products are to be stored.

Jarvis and Mcdowell (1991) prove that when the traversal (S-shape) 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 also show that this is only the case when the depot is located in the middle of the front of the warehouse and there is no demand correlation between products. Petersen and Schmenner (1999) evaluate commonly used storage policies like within-aisle, across aisle and diagonal aisle (figure 9). The within-aisle storage policy outperforms the other storage policies providing average tour lengths savings of 10-20%.

4.7. Order oriented slotting

It is noted that clever slotting depends for a great amount on the specific routing policy used (e.g. S-shape or largest gap). Surprisingly, researchers mentioned in 4.5 do not pay (much) attention to this issue. Mantel, Schuur and Heragu (2007) also consider the routing policy when dealing with slotting. They try to minimize the actual travel distance and name their approach "Order Oriented Slotting". Two parameters are introduced:

f_{i0} = popularity, the number of orders that require product *i*.

f_{ij} = interaction frequency, the number of orders that require both product *i* and *j*.

They argue that frequently jointly demanded product types (i.e., high interaction frequency) should be located close to each other, while popular product types (i.e., high order pick frequency) should not be located too far from the depot. In their analysis, they use the term distance to denote the *routing-specific-distance*. This way, they incorporate the routing policy. For example, for the S-shape policy, the distance denotes the number of aisles in between. Indeed, if an aisle contains a pick location, it must be traversed entirely. The distance between any two pick locations in the same aisle is of no importance in this case.
An integer linear programming model (ILP) is formulated for a single block warehouse where the S-shape routing strategy is applied (appendix III). The ILP model can only be solved to optimality for moderate size warehouses. Thus, two new heuristics are proposed for the slotting problem.
The *interaction frequency* heuristic is a constructive heuristic. First, *singles* (products that never share an order with other products) are allocated in accordance with their popularity. Thus, a frequently ordered single will be placed close to the I/O-point. Next, interaction frequencies are sorted in decreasing order and processed. The idea is that products *i* and *j* with a high interaction frequency f_{ij} should be placed close to each other and in accordance with their popularity. The authors do not provide a measure for the degree of popularity accordance.
The *interaction frequency based quadratic assignment heuristic* is a constructive heuristic as well. The authors formulate the slotting problem as a form of the quadratic assignment problem (QAP):

$$\min_{a\in S} z(a) = \sum_{i=1}^{I-1}\sum_{j=i+1}^{I} f_{ij} d_{ij}(a) + \alpha \sum_{i=1}^{I} f_{i0} d_{i0}(a) \quad (1)$$

In the expression above, S is the set of all storage assignments, $d_{ij}(a)$ denotes the routing-policy-specific distance (for storage assignment a) between product i and j and $d_{i0}(a)$ denotes the distance between product i and the I/O-point. The constant α provides relative weight to the objective of placing products demanded frequently near the I/O-point. The heuristic minimizes z over S. The authors argue that the value of α has to be determined empirically in order to obtain a correct balance between the two terms. Because the QAP is NP-hard, only small instances can be solved to optimality within a reasonable amount of time. For large instances (e.g. real-life situations), heuristics have to be used. Note that the value of z does not denote the actual travel distance. Indeed the routing-policy-specific distance between *any* product pair for which $f_{ij} \neq 0$ is enclosed in the expression. However, a reduction of the value z should also result in a reduction of the actual travel distance. Of course, after minimization of z, the actual travel distance has to be determined and compared to the current travel distance. As opposed to other authors, they use no statistical analysis to determine the correlation

between product types. Instead, the differences between f_{ij} values reveal a certain order pattern and thus provide directions for allocation. Finally, the authors provide a numerical example based on a *vertical lift module*. In this specific case, the value of α corresponds to 0, because the distance from a tray to the I/O-point (d_{i0}) is of no importance. Results show that the two heuristics perform significantly better than the commonly used COI-rule. However, no results are provided for cases in which $\alpha \neq 0$ (e.g. S-shape or largest gap).
Although Mantel, Schuur and Heragu (2007) provide a clear example of defining the routing-specific distance for the S-shape policy, it remains a difficult concept. Defining the routing-specific distance is not straightforward in case a warehouse has multiple blocks or when other routing strategies are applied (e.g. largest gap). No directions are provided in the article for these cases.

4.8. Direct link method

Many slotting methods described above identify how often product i and j occur on the same order. However, this does not necessarily means that an order picker collects product i before or after product j. In an attempt to model this issue, Van Oudheusden *et al.* (1988) developed a slotting method with the goal to minimize travel distance by simultaneously considering slotting and routing. It should be noted that the method is applied at a warehouse with aisle-captive AS/RS-equipment. The following notation is used:

$\tilde{d}_{ij}$ = Chebyshev distance (because of AS/RS-equipment).

$\tilde{f}_{ij}$ = direct link frequencies, e.g., how often is product i picked after or before product j.

The I/O-point is considered as a fictive product, which cannot be relocated. The procedure can now be summarized as follows:

0. Start with an initial lay-out and determine the values of d_{ij} and $\tilde{f}_{ij}$.
1. Compute $z = 1/2\sum_{i,j} \tilde{f}_{ij}\tilde{d}_{ij}$.
2. Compute $z' = 1/2\sum_{i,j} \tilde{f}_{ij}\tilde{d}_{ij}$.
3. If the value of z' can be reduced by interchanging the location of any two products, go to step 4. Otherwise go to step 6.
4. Interchange the locations.
5. Update the values of $\tilde{d}_{ij}$.
6. If z' equals z, go to step 9. Otherwise go to step 7.
7. Determine new picking tours.
8. Update the values of $\tilde{f}_{ij}$.
9. Stop.
10. The procedure can be restarted with another initial lay-out.

The solution procedure used is a hill-descending local search technique. A pair wise exchange procedure is used to swap products (step 3). Only improvements are accepted. Just like Mantel, Schuur and Heragu (2007), the problem formulation resembles a QAP. The value of z denotes the actual travel distance. A critical observation is that the contact frequencies f_{ij} between items i and j are not known in advance but stem from the solutions of the routing problems related to the set of order under consideration. The solution of the routing problems, however, is dependent on the location of the product types, which demonstrates the strong interrelationship between product location and routing.
Results show that the procedure reduces order picking time by approximately 13 % in a large Chinese warehouse. Van Oudheusden *et al.* argue that this method can also be applied to manual order picking

situations where multiple aisles are accessed by an order picker. Of course, the value of $\tilde{d}_{ij}$ must be changed accordingly. The authors do not indicate how this should be done.

4.9. Aisle congestion

Using a clever slotting strategy can result in a new problem: aisle congestion. This occurs when an aisle is filled with frequently demanded items and only 1 order picker at a time can access the aisle. In that case, there is a high probability that other order pickers have to wait leading to increased order picking time. Pan *et al.* (2005) apply queuing theory to a single block warehouse. They determine blocking probabilities (i.e., the probability that two or more order pickers want to access the same aisle). Next, an adapted COI rule is used thereby taking into account the blocking probabilities. They exemplify their approach with a fictitious small warehouse. Reductions in order picking time of about 22% are reported.

4.10. Concluding thoughts

Different models with respect to slotting have been proposed in literature with sometimes slightly deviating objectives. Slotting methods like the COI-rule or storage allocation patterns are easy to implement, but these methods wrongly assume single command orders and ignore the possible correlation between products. Still, reasonable results are obtained even when these strategies are applied in multi-command situations. Therefore, we implement the COI-rule and use the results as an upper bound. All modeled slotting problems for multi-command cycles or even deviated sub problems (e.g. the clustering problem) can not be solved in polynomial time. Obtaining optimal solutions for practical warehousing situations is therefore unlikely due to excessive required computation time.
Correlation measures between products are introduced by Frazelle (1989) and Amirhosseini and Sharp (1996). This way, they identify which products are likely to be ordered together. Subsequently, the authors argue that those products should be stored 'close' to each other in the warehouse. Correlation measures are useful for the slotting problem, but they only solve a part of the slotting problem. The researchers in this field ignore the allocation of clusters (placement of the clusters in a specific warehouse lay-out) or use rather simple solution methods for it (e.g. based on the COI-rule). This subtopic is, however, not a trivial part of the slotting problem (except in specific cases, see Garfinkel (2005)) and should not be treated as such. A critical observation concerns the relevance of an order profile. All researchers who deal with slotting use an order profile and assume this is relevant. Frazelle (1989) and Amirhosseini and Sharp (1996) do use correlation measures in order to determine probabilities that (two) products are ordered together. Those probabilities are still based on a given order profile. In case the order profile is *not* relevant, the effect of slotting will be minimal or even negative. However, in case of a relatively stable order pattern and a sufficiently large order profile, it can be safely assumed that an order profile *is* relevant. There is a strong interrelationship between optimal storage assignment and routing policies. Research that considers both topics is rather scarce and only concentrates on rather simple lay-outs (e.g. single block warehouses) and routing policies. Nonetheless, the research performed by Mantel, Schuur and Heragu (2007) seems valuable for our case. They explicitly consider minimization of travel distance. Although they concentrate on the S-shape routing policy for a single block warehouse, we feel that their research can be extended to other warehousing situations as well.
The direct link method (Van Oudheusden *et al*, 1988) is another interesting technique that considers routing and product allocation simultaneously and therefore we apply this method in our research. We do question, however, the performance of this heuristic in multi-aisles situations. Aisle congestion can be neglected in our case, because multiple order pickers can enter picking aisles simultaneously.

5 Selecting and adapting appropriate slotting methods

The current situation has been described and literature research has been performed. In this chapter, we deal with activities that have a major impact on order picking efficiency like routing and batching. Next, some specific issues concerning slotting at Wolters-Noordhoff are discussed. Then, heuristics for the slotting problem extracted from literature applied to Wolters-Noordhoff's situation are described. In addition, we design a new slotting technique. Incorporation of slotting strategies in our tool and obtained improvements are provided in the next chapter.

5.1. Subdivision of the manual picking area

In appendix I, a detailed overview of the manual picking area is provided. Part 1 includes aisle numbers 1 up to 81. Part 2 encloses aisle numbers 84 up to 93. Momentarily, elementary school products are located in part 2 of the picking area. Analysis of order data shows that orders containing these products are mostly demanded during a small part of the year. Furthermore, these products are not often demanded in combination with products belonging to other segments (e.g. high school or university products). The number of storage compartments in part 2 of the picking area more or less corresponds with the amount of elementary school products (taking into account a certain amount of free locations). Also, management is currently satisfied with the placement of this group of products in part 2 of the picking area. In addition, forecasts indicate that the demand pattern of elementary school products will not change significantly in the next few years. Because of these reasons, we decide to treat part 1 and 2 separately. This subdivision affects the batching and routing strategies implemented to a limited extent. We explain this in the next two sections and show that the resulting errors are small.

5.2. Splitting and batching of orders

As Petersen and Aase (2003) point out, batching has a major influence on travel distance. Thus, this process cannot be neglected. Order pickers at Wolters-Noordhoff use their own experience to batch and split orders. There are only two "hard" restrictions, which are described in section 3.2. Order pickers estimate how many orders can be added into one batch. This does not only depend on the total weight of the selected orders, but also on the quantities and sizes of the products. We investigated what rules of thumb are used for determining the maximum capacity of pick carts. Key observations are:

1. On average, about 50 order lines can be collected using a big pick cart. Big pick carts are only used for orders with a weight of more than 10 kg. Orders with a weight of at most 10 kg are collected with small pick carts. About 30 order lines can be collected using a small pick cart.
2. Most order pickers assume a maximum capacity of 100 kg for a big pick cart and 30 kg for a small pick cart.

Splitting of orders is performed if an order exceeds the maximum amount of order lines or the weight capacity of a pick cart. Of course, each part of the split order must satisfy the restrictions mentioned above as well. The original pick sequence is kept intact. Split orders are not batched with other orders. Next, we need to determine what rules are used to combine orders. Although not every order picker uses the same logic to combine orders, we observe that on average the following batching strategy is applied. First, a relatively large order (in terms of order lines) is chosen and added to the current batch. Then, another order is chosen such that the number of additional aisles, compared to the orders already in the batch, is minimized. Aisle numbers are displayed in appendix I. This process continues until the capacity of a pick cart is reached. Note that the capacity of a pick cart can be exceeded either by the total weight of the batch or the number of order lines.
We illustrate the splitting and batching process with an example. Consider the following set of orders:

Order	Weight per order	Number of order lines per order	Dispatch method
1	10 kg	10	TNT
2	50 kg	30	TNT
3	150 kg	10	CB
4	80 kg	150	CB
5	8 kg	5	TNT
6	40 kg	20	TNT
7	3 kg	2	TNT

Order 1,5 and 7 are small orders (in terms of weight). These orders are combined into one batch, because their total weight (10 + 8 + 3 = 21) and their number of order lines (10 + 5 + 2 = 17) do not exceed the capacity restrictions of a small pick cart (i.e., 30 kg and 30 order lines). Furthermore, the orders are all dispatched via TNT. Order 2 and 6 are batched, because they are both large orders (in terms of weight). Total weight (50 + 40 = 90 kg) and the amount of order lines (30 + 20 = 50) do not exceed the capacity restrictions of a large pick cart. The dispatch method for both orders is the same. Order 3 is split up, because its weight is more than 100 kg. Finally, order 4 is divided into smaller parts because of its amount of order lines (150). Of course, each part of the divided orders must also meet the restrictions. As we already mentioned, parts of divided large orders are not batched.
Besides the rules of thumb, the subdivision of the manual picking area (cf. 5.1) affects the results of the batching and splitting process as well. Orders which contain items located in both parts (part 1 and 2) are divided. We treat the resulting 'suborders' as individual orders. Thus, as an exception, these orders can be batched (only with orders that belong to the same part). Mainly elementary school products are located in the new part. Analysis of historical order data points out that orders with elementary school products seldom contain products of other segments. Only for a small amount of orders (about 2.5% of the historical order data used), both parts of the picking area need to be visited. Those orders are often large (both in terms of weight and order lines) and thus they are divided in smaller suborders. This often results in separate visits of the two parts. In conclusion, the resulting error of the subdivision will be very small.
Unfortunately, two other simplifications must be made in as well. Dispatch information of historical orders is not available. However, we do know that approximately 75% of the orders are shipped by TNT. Furthermore, large orders (in terms of weight) are often shipped by other companies. These orders are almost always split up and are not batched. Thus, this will hardly effect the outcome. The second simplification relates to rush orders. Every working day at 13.00, there is a chance that a small amount of rush orders is released to the work floor. In the available historical order data, no distinction is made between regular and rush orders. Thus, it is possible that a regular order is batched with a rush order. In practice, this will not occur. We expect that the impact of this error is small.
As we already mentioned, not all order pickers apply the same batching strategy. For example, some order pickers batch orders only by examining the first order line of each order. In addition, temporary workers need to break in and they often apply a form of random batching or no batching at all. We do not take these things into account. De Koster *et al.* (1999) show that the implemented batching strategy (minimizing additional aisles) performs well (in terms of distance reduction) in all kinds of warehouse lay-outs. Thus, the current travel distance determined by the tool may be in fact a underestimation of the real travel distance. The large degrees of freedom of the batching process necessitate assumptions and thus a very accurate estimate of the travel distance is not possible. By using the same assumptions for the current and new situation, we can still compare the outcomes and draw legitimate conclusions.

5.3. Routing policies

A vast amount of warehousing literature deals with the optimization of routing policies. A simulation study (Petersen and Aase, 2003) shows that optimized routing policies do not have such a great impact on travel distance in comparison to optimized batching and slotting strategies. At Wolters-Noordhoff order pickers are more or less 'guided' through the aisles by their scanning equipment. The mobile scanner sorts order lines on increasing location code and thus the scanner creates an order route. Order pickers still have the possibility to change the proposed picking route (by reshuffling location codes), but a large majority of order pickers and temporary workers does not use this possibility.

5.3.1 Assumptions with regard to routing

We assume that order pickers do not create routes on their own (i.e., the indicated pick sequence by the scanner is leading). The subdivision (cf. 5.1) implies separate order picking tours for part 1 and 2 of the manual picking area. In practice, orders which contain products of both parts, are collected in one order picking route. As we already mentioned in 5.2, only a small amount of orders contain products of both parts. Thus, the impact of this error will be minimal. The following assumptions are made with regard to part 1 of the picking area. First, we notice that order pickers often leave their pick cart in cross aisle 1 or 2 (figure 10) and then enter an aisle. They pick the required item(s) and return to their pick cart. Thus, the front and back aisles are often not used. We assume that the front and back aisles can not be entered and thus are never used. This is indicated by the bold lines in figure 10.

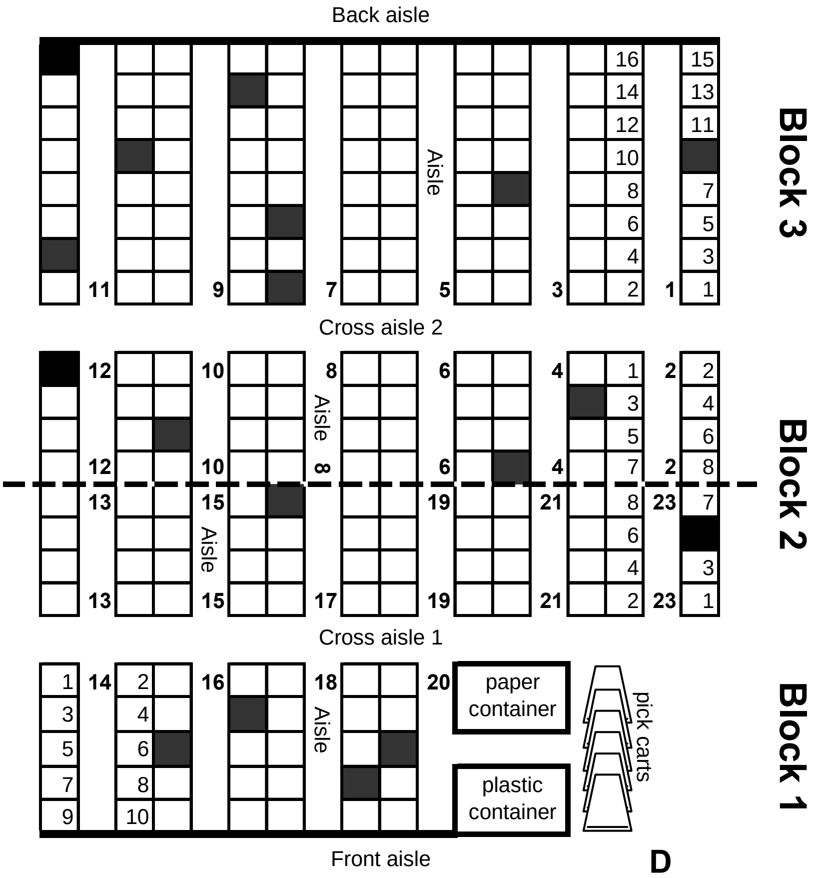


Figure 10: Overview of a small part of the manual picking area (part 1).

Secondly, we assume that the order picker chooses the minimal distance between two consecutive pick locations thereby taking into account the assumptions mentioned above. Thirdly, we assume narrow aisle picking. This means that an order picker can pick from both sides of an aisle without covering an additional distance. An order picker always walks in the centre of an aisle. An example is provided in the next section. Routing in part 2 is straightforward. In practice, order pickers often leave their cart in the front aisle, enter an aisle to pick the required items and return to their pick cart. Again, we assume that the back aisle is not used and narrow aisle picking is applied. The next section gives an example of both routing strategies.

5.3.2 Routing examples

To illustrate an order route in the manual picking area for part 1, consider figure 10. This figure represents a small section of the area. Aisle numbers are shown (in bold) as well as shelf locations (numbers in the squares). Because of readability, not all shelf location numbers are shown. The dotted line indicates that the centre block (2) actually consists of two aisles. The two cross aisles are also displayed. Although a front and back aisle exist, these paths may not be entered. This is indicated by the bold lines. Now consider the following (batch)order (compartment codes are replaced by XX, because they are not of interest for the routing strategy):

(Batch)order

Product	Location code
X1	01-09-XX
X2	03-08-XX
X3	04-04-XX
X4	04-07-XX
X5	07-02-XX
X6	07-06-XX
Y1	09-13-XX
Y2	10-05-XX
Y3	11-04-XX
Y4	11-08-XX
Y5	11-15-XX
Y6	12-01-XX
Z1	16-04-XX
Z2	16-05-XX
Z3	17-08-XX
Z4	18-08-XX
Z5	20-05-XX
Z6	23-05-XX

The order is already sorted on increasing location code. The constructed order route is displayed in figure 11. The route starts at the depot and the shortest distance (taking into account the assumptions mentioned above) between the depot and the first location is traversed. Then, the shortest distance between the current and the next location is traversed. If all pick locations have been visited, the order picker returns to the depot. Note that although item Z6 (location code 23-05-XX) occurs at the bottom of the pick list, it is actually the first picked item. By proceeding to aisle 1, aisle 23 is also entered. It seems logical that an order picker will pick the product(s) from aisle 23 straight away. In conclusion, the assumed pick sequence occurs according to increasing location code with the exception that the rightmost pick location is always visited first (regardless of the location code). This way, the resulting routing method is a combination of the midpoint and return strategy (Petersen, 1997).

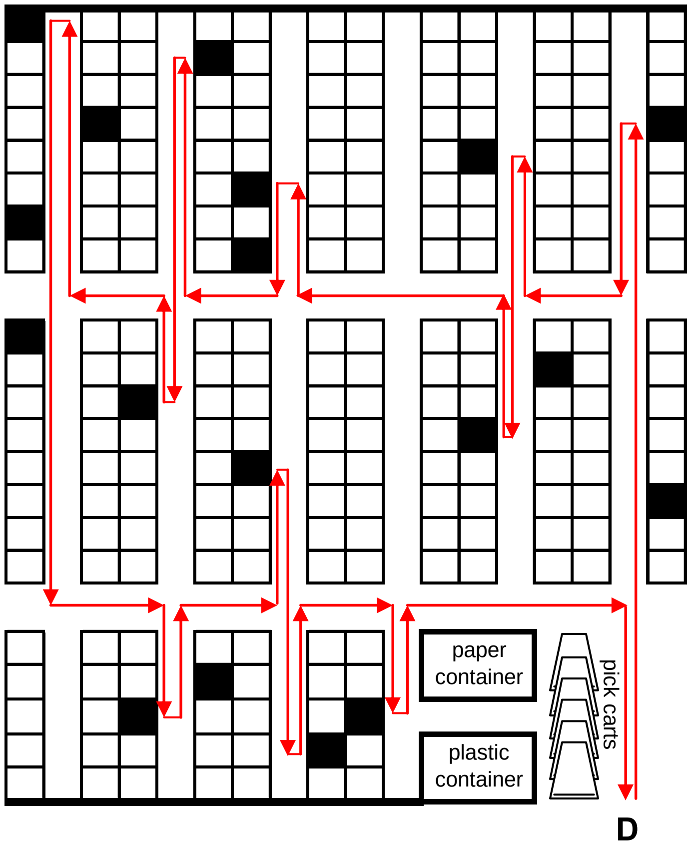


Figure 11: Example order picking route for part 1 of the manual picking area.

Order routes for part 2 are modeled as return routes. Figure 12 displays a section of the area. Aisle and shelf numbering are similar to that of part 1. Black squares indicate pick locations. Pick sequence is according to increasing location codes. No exception is made. The order picker starts and ends the route at the depot situated at part 1. The resulting route is according to a return strategy.

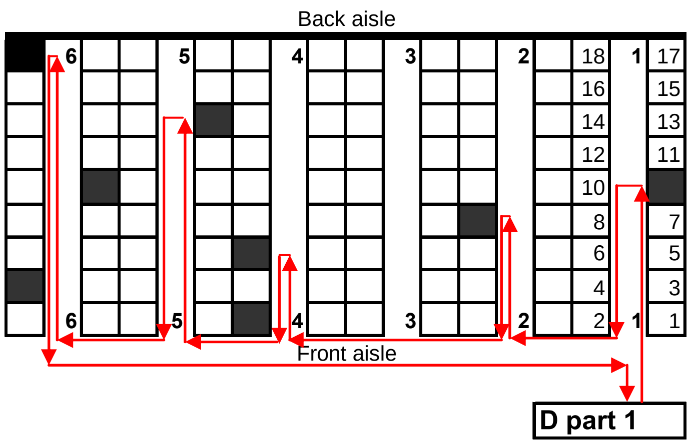


Figure 12: Example order picking route for part 2 of the manual picking area.

5.4. Product allocation issues

5.4.1 Compartment sizes and lay-out

Currently, dynamic pick locations are used in order to reduce travel time. There is a strong interrelationship between the allocated compartment sizes / capacities of the manual picking area and the amount of dynamic pick locations created. Frequently demanded products stored in small compartments (i.e., small capacities) will often be picked from the dynamic picking area. If those products are stored in larger compartments, dynamic pick locations will be created less often. Clearly, the decision on allocated compartment size has an impact on the total travel time. Although it is interesting to research the effect of optimally allocating compartment sizes to items, it is beyond the scope of this thesis. Thus, we assign the initially allocated compartment sizes (i.e., in the current situation) to products and consider compartment size as the volume of a product. So, a product initially stored in, for example, an A-compartment can *only* be moved to another A-compartment. This way, obtained outcomes can be related to the applied slotting strategy only (i.e., the original amount of dynamic picking locations does not change). In addition, this method always guarantees a feasible solution. Lay-out of a picking area influences order picking distance as well. At Wolters-Noordhoff, the aisle structure cannot be changed. This also holds for the size and amount of compartments per shelf (cf. appendix II for an overview of shelves and compartment types).

5.4.2 Occupancy of aisles

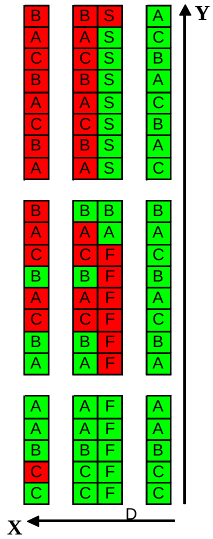
There are about 12,000 compartments in the manual picking area. On average, 8,800 product types are stored in this area. No specific statistical distribution concerning the division of free locations over the area can be given. Before optimizing the current allocation of products, we place products in such a manner that the occupancy of every aisle is roughly the same (i.e., a uniform distribution). The occupancy is calculated per part of the picking area and is as follows:

$$\text{Occupancy} = \frac{(\text{Number of compartments - number of products})}{\text{number of compartments}} .100\%$$

Note that we only take the *number* of free compartments into consideration. In other words, equal division of the *volume* of free compartments per aisle is not taken into account.

5.4.3 Product segment allocation

Wolters-Noordhoff is interested in the effect of a form of “zoning”. With zoning, we mean *in this case* that products belonging to the same segment must be placed as close together as possible. Due to the specific lay-out of the manual picking area, strict separation of segments is not feasible. Some picking aisles will contain products of different segments. A small example in figure 13 illustrates this.
An important question is how the five segments must be allocated. Indeed this affects the travel distance. Currently, elementary school products are placed in part 2 of the picking area. We do not reallocate this segment. Reasons are provided in section 5.1.



Segment 1	
#A-compartments	15
#B-compartments	14
#C-compartments	10
#S-compartments	7
#F-compartments	5

Segment 2	
#A-compartments	10
#B-compartments	7
#C-compartments	9
#S-compartments	1
#F-compartments	6

Figure 13: Two segments must be placed in the given warehouse lay-out. In this case, products are placed on increasing x coordinates. The depot is situated bottom right. Segment 1 is placed first. The letters in the lay-out (left) show the available compartment types. The tables on the right show which and how many compartment types are required per segment. Because of the given lay-out, no strict separated segments ("zones") can be created. This phenomenon also occurs if we apply "zoning" to the manual picking area.

The remaining four segments are placed in part 1 by using the following allocation rule (based on Liu, 1999). The total order frequencies of each segment are determined. Next, they are divided by the number of products per segment. We use the following notation:

F_x = average order frequency per segment
N_x = number of products belonging to segment x
f_{i0} = order frequency of product i

The average order frequency per segment is calculated as follows:

$$F_x = \frac{\sum_{i=1}^{N_x} f_{i0}}{N_x} \qquad i = 1.....N_x \quad \forall x = \text{segment } 1..4$$

The segment with the highest average order frequency is placed closest to the depot. Thereby, we also take into account the required equal occupancy per aisle (section 5.4.2). Optimal geometric placement of segments depends on certain lay-out parameters (De Koster and Le Duc, 2005). In this case we ignore these methods and use a more practical placement (after consultation with management). The geometrical placement of segments is displayed in figure 14. Note that zoning is an optional constraint. We provide results in chapter 6 with and without applying zoning.

Figure 14: Every colour corresponds to a certain segment. Elementary school products are placed in the upper part of the manual picking area. However, some elementary school products require certain compartment types and these can only be found in the lower part of the manual picking area. As a result, some aisles contain products of multiple segments. This issue also occurs for the other four segments. Thus, strict separation of segments is not possible.

5.5. Single Order Picking

At Wolters-Noordhoff, orders are either batched or divided in smaller parts before order picking. Minimization of additional aisles (section 5.2) is used to batch orders. Orders are split thereby leaving the initial pick sequence intact. These processes start from the current slotting of products. If we include batching and splitting of orders in the optimization techniques described below, we wrongly assume that certain products are frequently jointly demanded (because of batching) and other products are not (because of splitting). Therefore, we need to preserve the original order structure. Thus, we assume single order picking (every order is processed separately). After slotting optimization, batching and splitting is applied in the usual way in order to compare the new and old situation. Note that when batching is applied, improvements obtained with slotting will probably reduce. Extensive order batching implies longer order sequences that occur less frequently and consequently the effect of slotting diminishes. To give a rather extreme example, suppose we insert all orders of a given order set into one batch. As a result, we need to visit *every* pick location. In this case, slotting has no effect on the travel distance.

5.6. Demand pattern

Improvements obtained by a slotting strategy also depend on the demand pattern of products and changes in the product assortment. A very unstable demand pattern diminishes the effect of clever slotting. Indeed, product types often need to be removed (in order to minimize travel distance) incurring high costs of movement. Rapid changes in product assortment reduce the effect of slotting as well. We assume that the demand pattern of product types is rather stable for at least a couple of years. Some products are annually replaced by newer editions. It is expected that the demand pattern for these new editions does not differ significantly from previous editions.
We use order data extracted from the WMS for the incorporation of slotting methods in our tool and we assume this order profile represents the order pattern of Wolters-Noordhoff.

5.7. Storage assignment policies

In sections 5.7.1 up to 5.7.3 below, we adapt slotting strategies from literature to Wolters-Noordhoff's situation. In addition, we design a new slotting technique. Throughout the remainder of this chapter, the following notation is used:

Product types: $i = 1,2,3,....,I$
Orders: $k = 1,2,3,...,K$
Order set: $O_k \subset \{1,2,3,...,I\}$
Order frequency of product *i*: f_{i0}

Some heuristics use additional variables. These are stated in the accompanying sections.

5.7.1 Cube-per-Order-Index storage policy

This assignment policy is often applied in warehouses because of its practicability. Applying this method to the case of Wolters-Noordhoff is straightforward. Note that we consider the two parts of the manual picking area separately. In case of zoning, we apply this storage policy *per* segment. Segments are placed according to the rule described in 5.4.3. The heuristic is now as follows. First, distances between the depot and pick locations are measured in meters taking into account the aisle structure of the picking area. Then, order frequencies (f_{i0}) of all products are determined for the given order profile O and sorted in decreasing order. Distances from the I/O-point to each location (d_{i0}^m) are sorted in increasing order. The product with the highest order frequency is placed in a suitable location (e.g. the correct compartment size) nearest to the I/O-point. This process continues until all products are assigned to locations. Note that we neglect the volume of products in this case. After allocating all products, batching and splitting is applied to the given order set and the travel distance is determined.

5.7.2 Direct link method

We use the direct link method introduced by Van Oudheusden *et al.* (1988) and adapt it to our situation. Again, we treat part 1 and 2 of the picking area separately. A current storage layout and two matrices are needed for calculating cost reduction iteratively by pair wise interchanges of products. In the distance matrix $\mathbf{D}$, distances (in m) between all storage locations in the warehouse are recorded. Distances between the I/O-point and storage locations are also stored in this matrix. Van Oudheusden *et al.* (1988) use the Chebyshev distance, because in their case AS/RS-equipment is used. We use the minimal distance d_{ij}^m (in m) between two storage locations taking into account the aisle structure. Thus, for part 1 of the picking area, the front and back aisle (figure 10) cannot be entered. For part 2 (figure 12), the back aisle cannot be entered. Figure 15 provides some examples. Note that the distance matrix is symmetrical. The *direct link frequencies* $\tilde{f}_{ij}$ (how often is product *i* picked after or before *j*) are stored in a direct link matrix $\tilde{\mathbf{F}}_{ij}$. The I/O point is considered as a fictive product, which cannot be relocated. The direct link matrix is also symmetrical. For sake of clarity, we provide the steps of the heuristic again.

0. Start with an initial lay-out and determine the values of d_{ij}^m and $\tilde{f}_{ij}$.
1. Compute $z = \frac{1}{2}\sum_{i,j} \tilde{f}_{ij} d_{ij}^m$ (this is equal to the travel distance).
2. Compute $z' = \frac{1}{2}\sum_{i,j} \tilde{f}_{ij} d_{ij}^m$.
3. If the value of *z'* can be reduced by interchanging the location of two randomly chosen products, go to step 4. Otherwise go to step 6.
4. Interchange the locations.
5. Update the values of d_{ij}^m.
6. If *z'* equals *z*, go to step 9. Otherwise go to step 7.
7. Determine new picking tours.
8. Update the values of $\tilde{f}_{ij}$.
9. Stop.
10. The procedure can be restarted with another initial lay-out.

In step 3, only improvements are accepted (i.e., a reduction of *z*). After a swap of products *i* and *j*, the change in the objective value amounts to: $\sum_{x=0, x\neq i,j}^{I} (\tilde{f}_{jx} - \tilde{f}_{ix})(d_{ix}^m - d_{jx}^m)$.

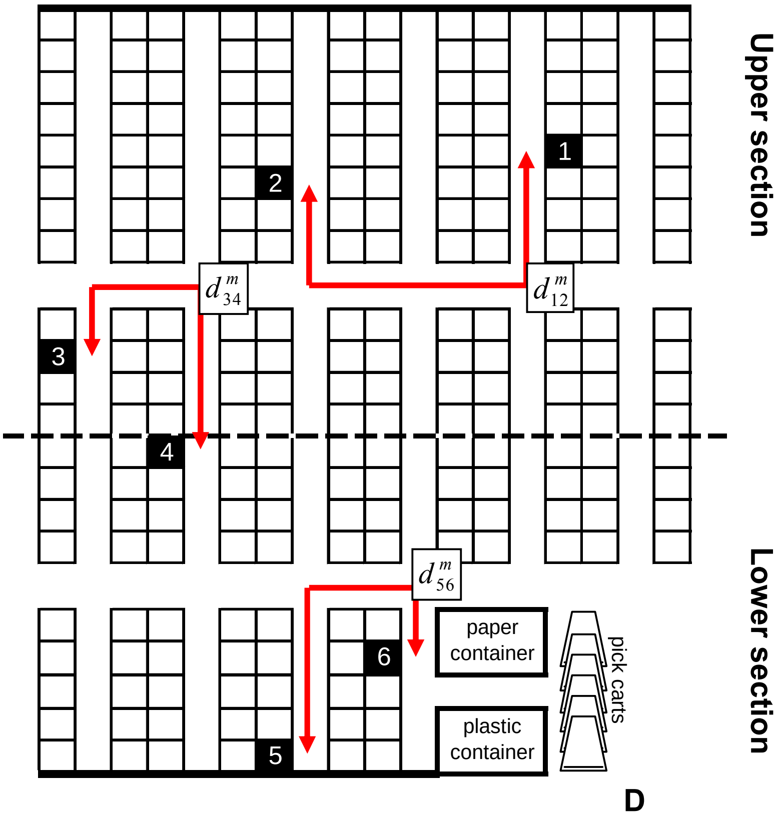


Figure 15: Examples of minimal distances between two pick locations ($d_{12}^m, d_{34}^m, d_{56}^m$) by taking into account the aisle structure.

Because a pair wise exchange procedure is used, the solution quality of the initial location pattern (step 0.) has an influence on the outcome of the heuristic. Therefore, starting from a random allocation of products is not sensible. We come back to this issue in chapter 6. Van Oudheusden *et al.* apply pair wise interchanges in step 3 until no change could be made resulting in a profit of more than 0.05% of the current value of the objective function z. In the next chapter, we adjust this value. After termination of the heuristic, the batching and splitting process is executed and the travel distance is determined.

5.7.3 Order oriented slotting methods

5.7.3.1 Order oriented product swapping

Mantel, Schuur and Heragu (2007) introduce the concept of order oriented slotting. Methods like the COI-rule and storage allocation patterns are product oriented. The order structure is neglected in those cases. The authors have developed slotting techniques that do take order structure into account (section 4.7). We have designed a slotting heuristic that also focuses on order structure. The heuristic can be summarized as follows.

1. Given an initial layout, determine the picking tours for a given order profile O and calculate the total travel distance z.
2. Choose two products i and j randomly. Create a subset S in which orders can be stored.
3. Determine which orders contain exactly *one* of the two products (i or j) and store those orders in subset S.
4. Calculate the travel distance for the orders in subset S.
5. Swap the locations of products i and j.
6. Determine picking tours for the orders in subset S and calculate the accompanying (altered) travel distance.
7. Calculate the difference in travel distance between step 4 and 6. A reduction in travel distance is always accepted.
8. Iterate steps 2 up to 7 until a stopping criterion has been reached.
9. STOP.

Note that we perform the heuristic per part of the picking area. Batching and splitting of orders is executed after termination of the heuristic. It is not necessary to determine picking tours for orders which contain both products i and j (step 3). Indeed, those routes do not change after a swap. The only difference is that either product i or j is collected first instead of the other way around. Reductions in travel distance are always accepted (step 7). If we *only* accept reductions in travel distance, we always end up in the same local minimum. In order to avoid getting trapped in local minima, we apply simulated annealing to the slotting problem. The concept of simulated annealing is based on the physical annealing process of solids. It is a combinatorial optimization technique in which improvements (in this case a reduction in travel distance) are always accepted, while deteriorations are also accepted to a limited extent. In appendix IV, simulated annealing is explained in detail. The computation time of this heuristic depends for a large amount on the size of the order profile. Determining which orders contain either product i or j will take more time as the amount of orders increases. The amount of orders that contain either product i or j will probably increase. Therefore, determining picking routes and calculating the accompanying travel distance will also require more computation time. In case of many large orders and advanced routing policies (e.g. the optimal), this heuristic requires far too much computation time. However, at Wolters-Noordhoff, the average order size (in terms of order lines) is small and determining picking routes can be done quite efficiently. We explain this in the following chapter.

5.7.3.2 Interaction frequency heuristic

This heuristic (Mantel, Schuur and Heragu, 2007) deals with the order structure by exploring interaction frequencies between pairs of products. As we already explained in chapter 4, the interaction frequency between product *i* and *j* denotes how often these products occur on one order. Thus, a simplified order structure is used. Again, part 1 and 2 of the picking area are dealt with separately. The heuristic consists of the following steps.

0. Determine the order frequencies f_{i0} and interaction frequencies f_{ij} of all products (I). The order frequencies are recorded in a 1xI-matrix $\mathbf{F_{i0}}$. The interaction frequencies are stored in a symmetrical IxI-matrix $\mathbf{F_{ij}}$.
1. Determine the *routing-specific* distances (taking into account the assumptions concerning the routing policies) between any two pick locations d_{ij}^{r} (including the I/O-point, d_{i0}^{r}) and store these values in a distance matrix **D**.
2. Determine the locations of all products in case a COI storage assignment is applied (section 5.7.1). Call these tentative locations *COI-locations*.
3. Check which products are never ordered together with other products (e.g. the *singles*). Assign these products to their *COI-location*.
4. Sort the non-negative interaction frequencies (>0) in decreasing order.
5. Consider a certain f_{ij}. If products *i* and *j* have already been allocated, then process the next interaction frequency. Otherwise, two situations can occur:
   a. *Products i and j have not been allocated*. Create for product *i* a set A_i and for product *j* a set A_j in which *allowed locations* can be stored. Add the *COI*-location of product *i* to set A_i and the *COI-location* of product *j* to set A_j. Check the distances from the *COI-location* to the I/O-point for both products (d_{i0} and d_{j0}). Consider set A_i. Given a certain factor β (>0 and <<1), a free location *x* for which holds that $(1-\beta)d_{i0}^{r} \leq d_{x0}^{r} \leq (1+\beta)d_{i0}^{r}$ is added to set A_i. Naturally, only locations with suitable compartment sizes are considered. After set A_i has been filled with allowed locations, the same procedure is executed for set A_j. Then, we choose the locations of product *i* and *j* from the sets A_i and A_j such that d_{ij}^{r} is minimal.
   b. *Either product i or product j has been allocated.* We apply the same procedure as in a., but now the location of one product is already fixed. Thus, a set of allowed locations for only one product needs to be created. If no suitable location can be found, process the next interaction frequency.
6. After processing all f_{ij}'s, some products remain unassigned, because their allowed locations are already occupied by other products. The order frequencies of unassigned products are sorted in decreasing order and free locations are determined. The remaining unassigned products are allocated based on the COI storage policy.

As we already stated in section 4.10, the *routing-specific-distance* d_{ij}^{r} is a difficult concept. In case the S-shape routing policy is used and a warehouse lay-out consists of 1 block, d_{ij}^{r} equals the difference in aisles between the location of product *i* and *j*. Indeed, if an aisle needs to be visited, it is traversed entirely. It is possible to formulate the problem as an integer linear programming problem for that specific situation. Appendix III states this formulation. A numerical example is provided as well. At the warehouse of Wolters-Noordhoff, different routing policies are applied and the warehouse lay-out consists of multiple blocks (figure 10). Defining values of d_{ij}^{r} for part 1 of the picking area is not always straightforward. If two products are located in the same section (upper or lower section, cf. figure 17), only one route is possible and d_{ij}^{r} represents the minimal distance between the two locations (taking into account the routing assumptions for part 1). This is not true if two products are located in different sections of the picking area. The example in figure 17 shows that for a certain d_{ij}^{r} more than 1 value is possible depending on the placement of other pick locations (as opposed to a single block warehouse with a S-shape routing policy). We tried to deal with this issue by identifying the probabilities that a certain picking path is chosen. We performed some small experiments to investigate the usefulness of these probabilities. Unfortunately, the outcomes of the experiments were not satisfactorily. Therefore, we ignore this issue and denote the distance displayed in situation B as the *routing-specific-distance* d_{ij}^{r}. Thus, we use the minimal distances between all product locations taking into account the previously mentioned assumptions (e.g. front and back aisles cannot be accessed). Defining routing-specific distances for part 2 of the picking area is not difficult. This part consists of 1 block and a return routing policy is applied. Thus, per product pair *i* and *j*, there is only one route possible (i.e., $d_{ij}^{m} = d_{ij}^{r}$). Figure 16 provides an example.

The value of β depends on the layout of the warehouse and needs to be determined empirically (chapter 6). Products that are often jointly requested should intuitively be placed closed to each other. However, frequently ordered products should not be placed too far from their COI-location. The heuristic tries to combine these two ideas.

5.7.3.3 Interaction frequency based quadratic assignment heuristic

Mantel, Schuur and Heragu (2007) formulate the slotting problem as a form of the quadratic assignment problem (section 4.7). The problem formulation is as follows:

$$\min_{a \in S} z(a) = \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} f_{ij} d_{ij}^{r}(a) + \alpha \sum_{i=1}^{I} f_{i0} d_{i0}^{r}(a) \qquad (1)$$

Recall that f_{i0} denotes the order frequency of product *i* and f_{ij} denotes the interaction frequency of product *i* and *j*. In addition, $d_{i0}^{r}(a)$ denotes the *routing-specific* distance between product *i* and the depot and $d_{ij}^{r}(a)$ equals the *routing-specific* distance between product *i* and *j* (for storage assignment *a*). Finally, the constant α indicates the relative weight of the term $\sum_{i=1}^{I} f_{i0} d_{i0}^{r}(a)$. This term forces frequently ordered products to be placed near the depot. The heuristic places products that are frequently ordered close together while fast movers are allocated close to the depot by minimizing the value *z*. The parameters f_{i0}, f_{ij} and d_{i0}^{r} can be determined rather easily (chapter 6) for a given order profile *O* and a warehouse layout. However, determining the values of d_{ij}^{r} is not straightforward. We already discussed this issue in section 5.7.3.2. In this case, we set d_{ij}^{r} equal to d_{ij}^{m} for both parts of the manual picking area. The authors argue that the value of α needs to be determined empirically. We introduce the following rule of thumb for the determination of α:

$$\alpha = \frac{\sum_{i=1}^{I} \sum_{j=i+1}^{I} f_{ij} d_{ij}^{r}(a)}{\sum_{i=1}^{I} f_{i0} d_{i0}^{r}(a)} \qquad (2)$$

Thus, by using the rule of thumb the weight of the two terms will be initially the same. Again, a simple pair wise exchange method (as in section 5.7.2) can be used for minimizing this quadratic assignment problem (QAP). However, many optimization techniques exist that obtain better results. In appendix V, we give a short introduction to these methods and select the most appropriate one for our case. The core of these methods is still based on a two-exchange procedure.

After a swap of product *i* and *j*, the difference in the objective function (1) amounts to:

$$\alpha(f_{i0}-f_{j0})(d^{r}_{j0}(a)-d^{r}_{i0}(a))+\sum_{x=1,x\neq i,j}^{I}(f_{jx}-f_{ix})(d^{r}_{ix}(a)-d^{r}_{jx}(a)) \qquad (3)$$

By minimizing the value *z*, the two terms will get different values. Thus, after each swap, the value of α needs to be adjusted in order to obtain equal weights of both terms. Because this adjustment is rather time-consuming, we adjust the value after a certain amount of swaps (chapter 6). The problem formulation above does *not* represent the actual traversed distance (this is also true in case of a single block warehouse with S-shape routing policy, cf. appendix III). Indeed, for a certain order *k*, distances between all product pairs of order *k* are enclosed in the objective function. The following example explains this. An order *k* consists of three products: *i,j* and *l*. The pick sequence for this order is as follows: *i* → *j* → *l*. Thus, only the distances between product pair *{i,j}* and product pair *{j,l}* are traversed (distances from and to the depot are omitted). However, interaction frequencies are: f_{ij} =1, f_{il} = 1 and f_{jl} = 1. Distances between the three products are non-zero. As a result, the first term of the objective function does not represent the actual distance. The same line of reasoning holds for the second term. Distances from each location to the depot are enclosed in objective function (1), whereas in reality an order picker only travels from the depot to the *first* pick location and from the *last* pick location back to the depot for any given order. Although the traversed distance is not represented in the function, a minimization of the objective value should also result in a minimization of the actual traversed distance.

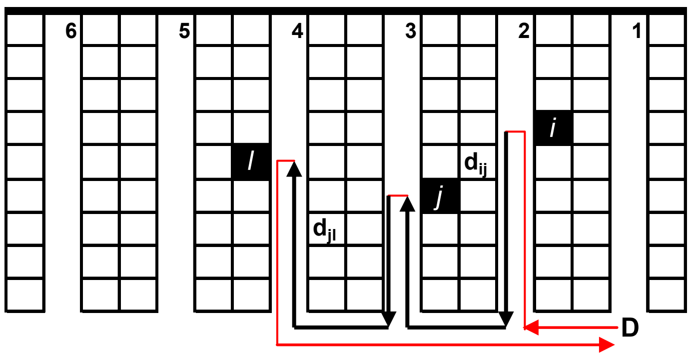


Figure 16: This example relates to part 2 of the picking area. Black squares indicate pick locations. Pick locations are sorted on increasing location code. Routing-specific distances are straightforward in this case, because the lay-out consists of only one block and a return routing policy is applied. Routing-specific distances d^{r}_{ij} and d^{r}_{jl} are displayed as black lines.

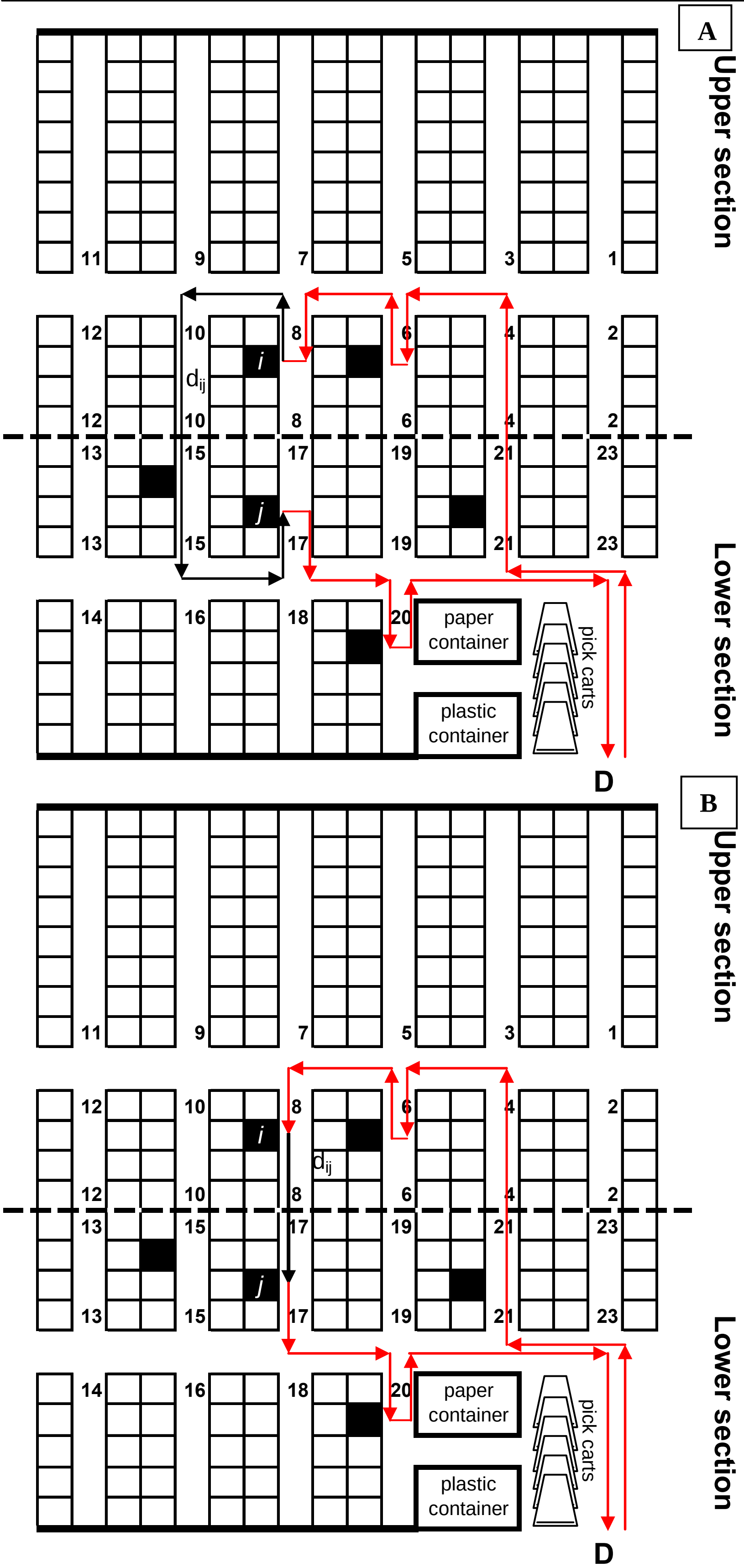


Figure 17: Black squares indicate pick locations. First, the right-most located product is picked. The picking order of the remaining products is according to increasing location code. The resulting routes are shown. The routing distance between product i and j differs between the two situations (A and B). The routing distance d_{ij}^{r} (shown as a black line in both figures) depends on the position of the left-most pick location in this example.

5.8. Conclusions

In chapter 4, we performed literature research. We observed that many researchers deal with identifying “closeness” relationships between products (i.e., product pairs (Frazelle, 1989) or product clusters (Rosenwein, 1994)). Many of those methods do not explicitly work with travel distance. Instead, they use one of several surrogate measures of cluster strength. Goal of our research is to determine one or more slotting strategies that minimize travel distance in the manual picking area at Wolters-Noordhoff. Routing policies affect the efficiency of slotting. The slotting methods described in this chapter take this into account as opposed to many other storage policies. Although batching influences slotting as well, we do not consider batching and slotting simultaneously. We do observe that extensive batching diminishes the effect of clever slotting. In the next chapter, we incorporate these methods in our slotting tool, compare their efficiency and select the most appropriate slotting method.

6 Construction of a comparative slotting tool

In the previous chapter, we discussed five slotting methods. This chapter describes the incorporation of these methods in our slotting tool. The process of (order) data gathering is also discussed. The results of some slotting methods depend on the fine-tuning of certain parameters, which is also addressed in this chapter. Based on results provided, we select the most appropriate slotting method for Wolters-Noordhoff.

6.1. Functionality of the slotting tool

The slotting tool will be used by the warehouse manager. It must be designed in such a way that the following questions can be answered:

1. Which slotting method provides the largest reduction in travel distance for a given relevant order profile?
2. What is the influence of zoning (section 5.4.3) on the results?
3. How can the products be reallocated?

The required steps in order to answer these questions are as follows. First, we determine the traversed distance for a given order profile *O* based on the current allocation of products, the assumed routing policies, batching and splitting methods. The current distance is displayed in the tool. Then, we start with the optimization phase. During this phase, we assume all orders of order profile *O* are picked on individual basis (section 5.5). However, orders which contain products located in both parts of the picking area are split up, because we treat part 1 and 2 of the picking area separately. Analysis of order data shows that those orders do not very often occur. First, products are (randomly) reallocated per part of the picking area in such a way that the occupancy of each aisle is roughly the same (section 5.4.2). Naturally, the initially assigned compartment type of a product (appendix II) may not be altered (section 5.4.1). Then, the five slotting methods are applied to part 1 and 2 of the picking area (sections 6.5 - 6.9). Unoccupied locations are not considered during execution of these methods. Next, batching and splitting is executed based on the new allocation of products. Finally, the traversed distance for the given order profile *O* and the new allocation of products is determined and displayed in the tool. The accompanying reduction in travel distance is displayed as well. The tool also creates simple reallocation assignments. These assignments are meant for order pickers and could be incorporated in the WMS. Figure 18 provides an example.

Product	Current location code	New location code
A	1	3
B	3	4
C	6	2
D	4	1
E	5	6
F	2	5
G	8	9
H	9	7
I	7	10
J	10	8

Reallocation ass.	Products
1	A - B - D
2	C - F - E
3	G - H - I - J

Figure 18: Example of reallocation assignments. For ten products (A up to J), the current and new location codes are provided. Every colour represents a reallocation assignment. Products are moved one by one by an order picker using a pick cart. We now explain how reallocation assignment 1 is created. Product A is stored on a pick cart and the order picker travels to the new location of product A (3). The initial location of A (1) is now unassigned. Location 3 is currently occupied by product B. The order picker puts product B on its pick cart and stores A in its new location (3). Then he travels to the new location of product B (4). Location 4 is currently occupied by product D. He puts product D on its pick cart and stores B in its new location (4). Then he travels to the new location of product D (1). Location 1 is unassigned and he can store product D in location 1. His pick cart is now empty. As a result, a reallocation assignment is finished. The same line of reasoning can be applied to reallocation assignments 2 and 3.

In case of zoning (zoning is an option), we use a slightly different approach for the optimization phase. First, elementary school products are located in part 2 to the extent that suitable compartment types are available. Those products are randomly allocated in such a way that the occupancy of each aisle is roughly the same. We determine the average order frequencies F_x for the other four segments located in part 1 and place the segment with the highest average order frequency closest to the depot (section 5.4.3). Again, products are placed such that the occupancy of each aisle is roughly the same. The remaining steps are identical to the case zoning is not applied. An overview of the optimization phase is given in figure 19. Implementation issues concerning routing, batching and splitting are discussed in the next two sections. We used Microsoft Access 2003 in combination with Visual Basic for Applications to program the tool as required by management of Wolters-Noordhoff.

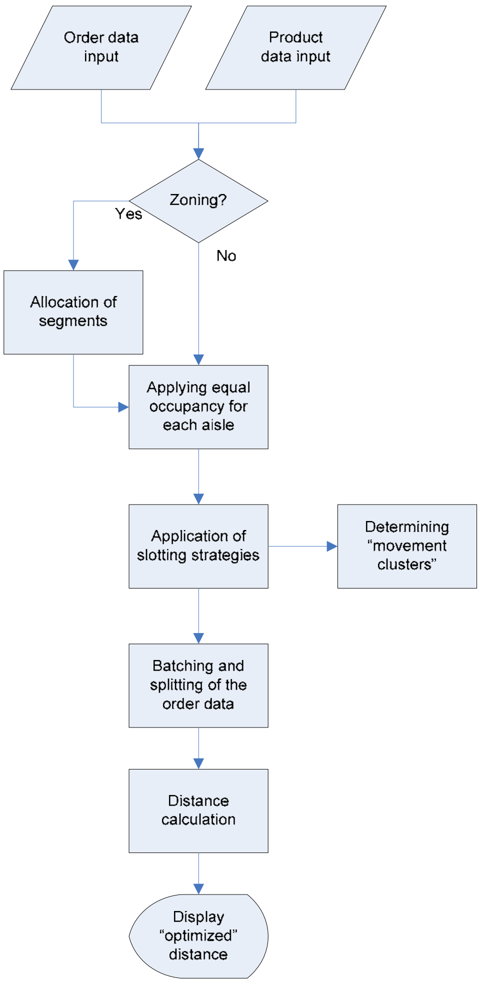


Figure 19: Overview of the optimization phase of the slotting tool. Note that these steps are executed per part (1 or 2) of the picking area.

6.2. Routing policies

Part 1 of the manual picking area is placed in a two-dimensional grid. We assign a position number to each pair of opposite located shelves. Figure 20 provides an example. Position numbers are linked to a (x,y)-coordinate in the following way:

L = length of the picking area expressed in amount of position numbers (= 21 in this case).

y-coordinate position number i = ((i -1) Mod L) + 1.
x-coordinate position number i = $\lfloor (i-1)/L \rfloor + 1$.

For example, the coordinate of position number 80 equals (4,17). The position number of the depot equals 0. Next, we determine the dimensions of the warehouse. These can be found in appendix I. Then, we calculate minimal distances between all position numbers given the coordinates, the dimensions of the warehouse and the various routing assumptions we made (section 5.3.1). The distances between pairs of position numbers (the depot included) are stored in a matrix **D**. Note that this matrix is symmetrical.

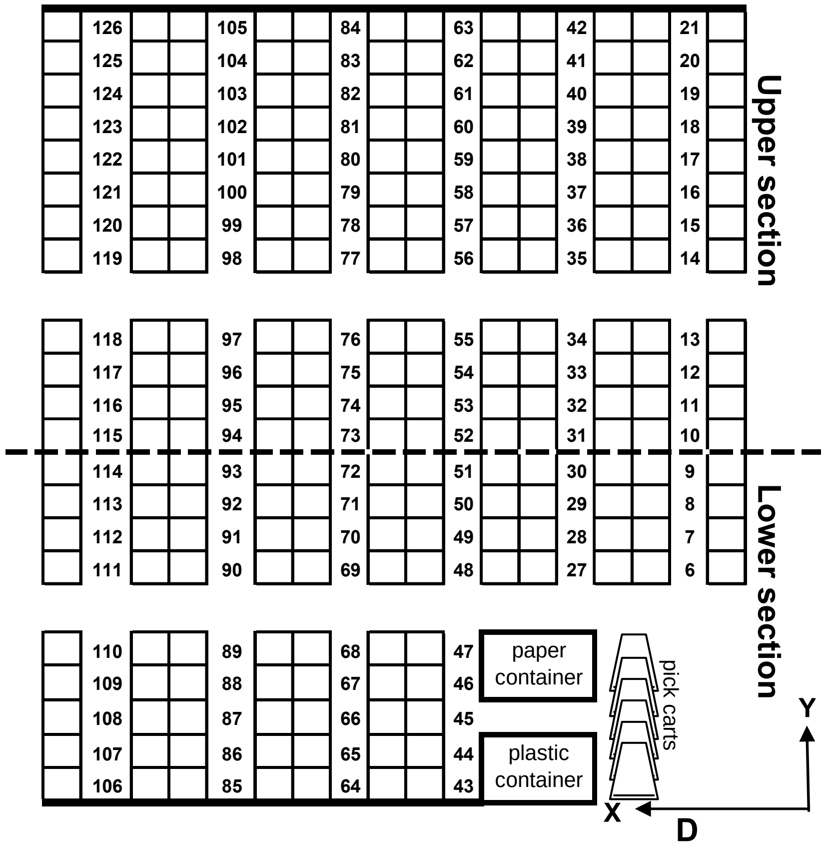


Figure 20: Example of location numbering for part 1 of the manual picking area. Only a small part of the manual picking area is displayed.

The routing distance for a given order can now be determined. First, position numbers of the required products are sorted. Numbers of the upper section are sorted in increasing order and numbers of the lower section in decreasing order. Based on this sorting process, the distance for a given order is simply determined by summing up all distances between two consecutive position numbers. Because the average amount of position numbers per order is rather small, we use bubble sort. The same methodology is used for part 2 of the picking area (figure 21). Ranking position numbers in increasing order for a given customer order results in the correct return routing strategy.

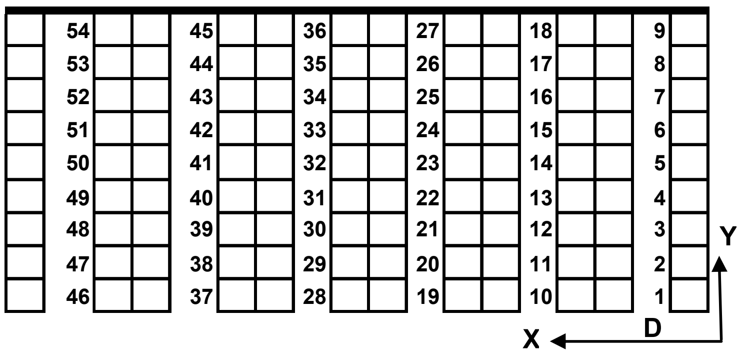


Figure 21: Position numbering of part 2 of the manual picking area. Position numbers are sorted increasingly. This results in a return routing policy.

The symmetrical distance matrix **D** (both for part 1 and 2) needs to be initialized only once. This requires very little computation time (< 1 s). Sorting position numbers per order also requires little CPU time, so determining order routes can be done very efficiently.

6.3. Batching and splitting process

In section 5.2 we explained which assumptions we made for the batching and splitting process. We investigated which rules of thumb order pickers use. In the slotting tool, the warehouse manager can adjust these rules of thumb. He can increase or decrease the weight capacity of pick carts. The maximum number of order lines per batch can be adjusted as well. Naturally, adjusting these parameters influences the effect of slotting. As the allowed size of batches decreases, the effect of slotting increases and vice versa. Results provided in the following sections are based on our rules of thumb for the batching and splitting process. Based on the flowchart provided in figure 22 we incorporate this process in our slotting tool.

6.4. Order and product data collection

We use order data from the months July up to November of 2006. This way, both ‘peak’ and ‘normal’ months are enclosed in the data. We assume this order profile is relevant for the order pattern of Wolters-Noordhoff. The Warehouse Management System provides us with the required order and product data. Each order line consists of a(n):

1. Invoice date.
2. Invoice number.
3. Required product code.
4. Required amount of items.

The following data is available for each product:

1. Product code.
2. Product description.
3. Product segment (i.e., elementary school, high school, university, professional education or additional educational products).
4. Location code (section 3.3).
5. Weight (in grams).
6. Currently assigned compartment code (A,B,C,F,K,L,M,N,S or W, cf. appendix II).
7. Compartment capacity for the product (expressed in item quantity).

No distinction in the order data is made between the dynamic, manual and bulk picking locations. However, order picking from bulk locations and dynamic locations is not relevant for our research. Therefore, we need to identify which order lines are picked from the manual picking area. Recall that on a given working day a dynamic picking location for a product is created if the total requested item quantity of a product exceeds the accompanying compartment capacity in the manual picking area. Because we do know the required amount of items per order line and the compartment capacities of products, we can delete the order lines that are picked from dynamic locations. In case a product is requested in pallet quantity, it is picked from the bulk area. This quantity is not taken into account when determining whether a dynamic picking location should be created or not. Thus, we need to know the pallet quantities of each product. Otherwise, we wrongly assume that certain order lines are not picked from the manual picking area. Unfortunately, this information is not available and therefore we estimate pallet quantities. We assume that a pallet quantity of product i is equal to the compartment capacity of product i times 4. We feel that the error resulting from estimating pallet quantities is smaller than the error resulting from simply neglecting bulk picking.

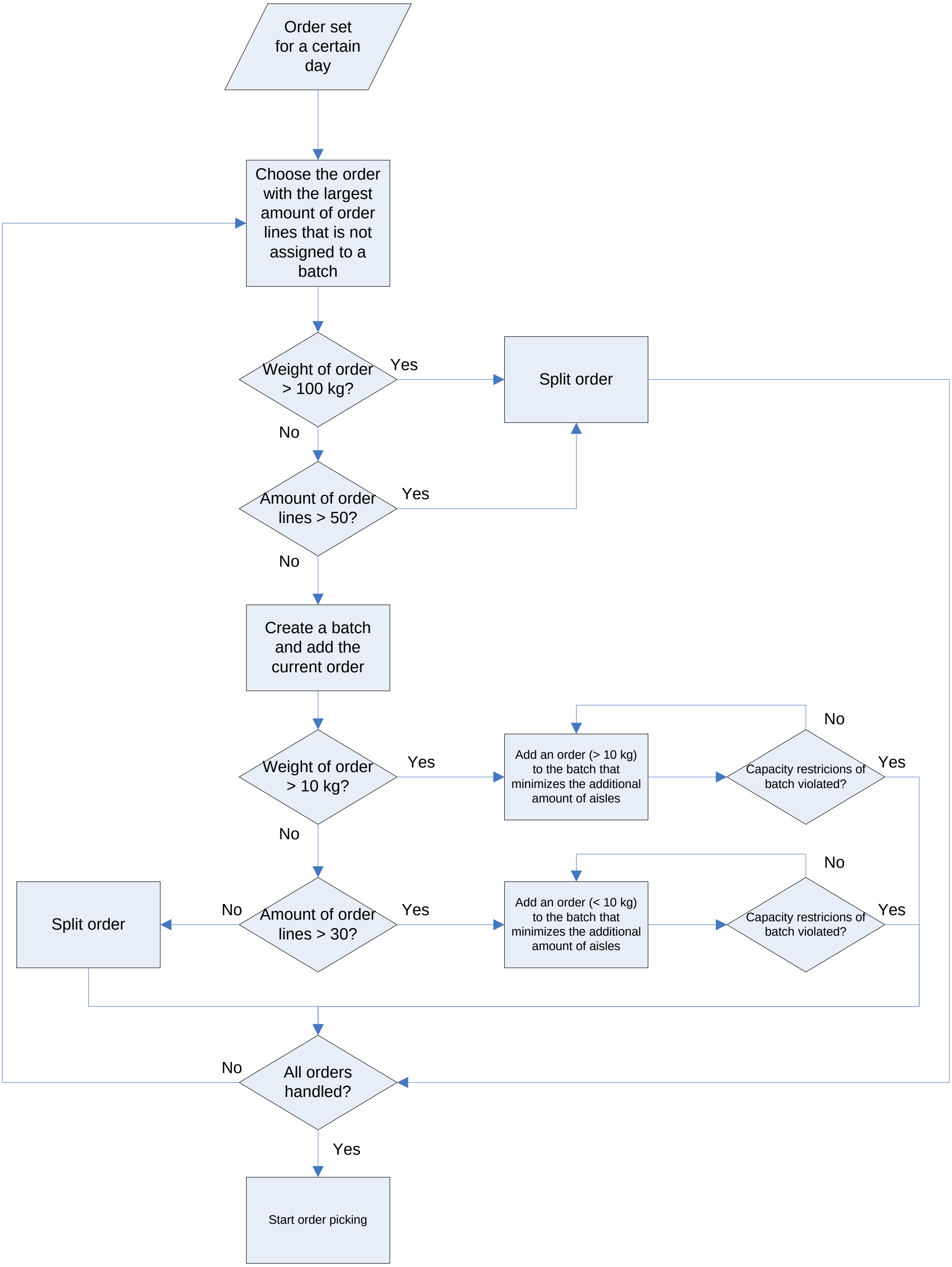


Figure 22: Flowchart of the batching and splitting process. Naturally, only orders with the same invoice date can be batched. Standard rules of thumb are used. The warehouse manager can adjust these rules in the slotting tool (i.e., the weight capacity for pick carts and the maximum number of order lines per batch).

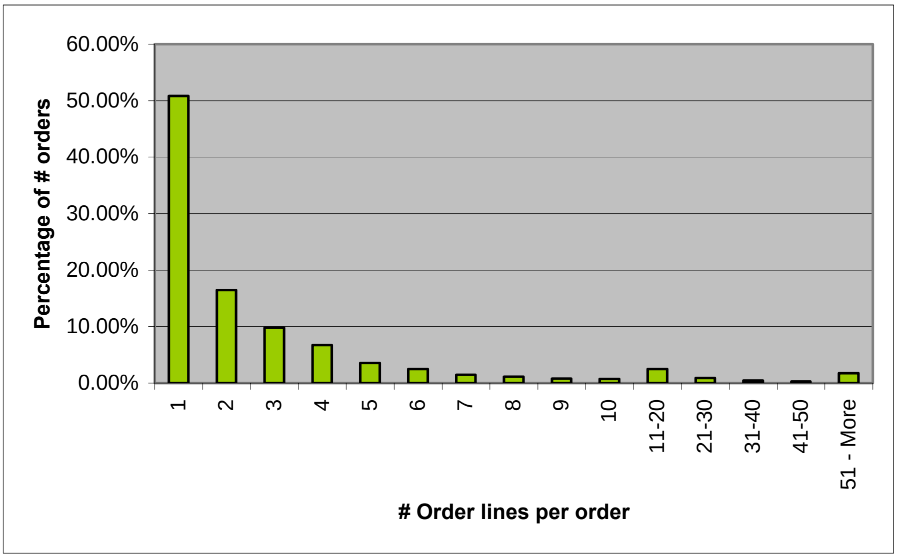


Figure 23: Histogram of the number of order lines per order.

We explain the 'cleaning' process of data by the following example. Suppose 4 orders need to be picked. Each of these orders contains product *i*. Item quantities per order for product *i* are as follows:

Order 1		Order 2		Order 3		Order 4	
product	quantity	product	quantity	product	quantity	product	quantity
i	10	*i*	20	*i*	200	*i*	10

Inventory data for product *i* is stated below.

product	capacity manual picking area (actual data)	<u>*estimated*</u> pallet quantity (actual data is not available)
i	50	4. 50 = 200

The requested amount of items equals 240. In case we neglect bulk picking, we would assume that all orders are picked from the dynamic picking area. Indeed, the amount of requested items exceeds the capacity of the manual picking area. Consequently, order lines from orders 1 up to 4 would be deleted. However, based on the estimated pallet quantity, it can be seen that order 3 is picked from bulk. The remaining item quantity equals 40. So, orders 1,2 and 4 are in fact picked from the manual picking area. Concluding, we only delete the order line from order 3.
Data is recorded in a Microsoft Access Database. Inserted order data is automatically 'cleaned' using the procedures described above. We used queries for this process. The resulting order profile encloses more than 40,000 orders. The amount of order lines picked from the manual picking area amounts to approximately 275,000. The distribution of the order sizes in terms of order lines is displayed in figure 23. Note that a very large part (> 50%) of the order profile consists of so-called *single-line orders*. The number of products currently stored in the manual picking area amounts to 8,800. Analysis shows that approximately 25% of the products covers almost 80% of all order lines (for the given order profile). In addition, almost 50% of the products covers only 5% of the order lines. This effect is not uncommon in warehouses.
In the database we link location codes to position numbers (section 6.2). Note that a location code in fact represents a compartment (section 3.3). Every shelf consists of several compartments (appendix II). Thus, several location codes can have the same position number. Compartment codes (A,B,C, etc) are linked to location codes and to products based on the *current* storage assignment. When a product is reallocated to a new location code, the compartment code of that new location code and the compartment code assigned to the product must match. So, a product that is currently stored in compartment type B can *only* be moved to other B-compartments.

6.5. Results of the COI slotting method

For this policy, we basically need to determine the order frequencies f_{io} and the distances from each position number to the depot. The latter are stored in distance matrix **D** (section 6.2). Order frequencies of all products (*I*) are stored in matrix $\mathbf{F_{i0}}$. Then, we apply the methodology described in section 5.7.1. Both matrices are sorted (in increasing respectively decreasing order) using shell sort. In order to see what the impact is of the batching and splitting process on slotting, we also provide results (table 1) for the single order picking case (SOP). In addition, results with and without zoning are provided for each case. Computation times (only for the slotting part and allocation of segments in case of zoning) are provided as well. We used a Pentium IV 2Ghz with 256 MB of RAM.

Results SOP	Distance (km)	Reduction (%)	CPU time (hh:mm:ss)
Current distance	5292	-	00:00:04
COI without zoning	3855	27.2%	00:00:22
COI with zoning	3923	25.9%	00:00:30

Results after batching and splitting	Distance (km)	Reduction (%)
Current distance	2029	-
COI without zoning	1816	10.5%
COI with zoning	1853	8.7%

Table 1: Results of the COI slotting method. Total travel distances (i.e., for part 1 and 2 together) are displayed.

We expect that this heuristic would perform poorly, because a single-command order picking system is assumed and order structure is neglected. Results prove otherwise, especially in the case of single order picking (SOP). This is probably due to the structure of the order profile and could be explained as follows. Possibly, products that are often ordered together more or less have the same order frequency. In addition, the variation of order frequencies is high. As a result, COI will place these items not very far from each other.

After the batching and splitting process, results deteriorate as we expected. Note that the results with and without zoning only slightly differ. This can be explained as follows. In case of zoning, products of the same segment are placed relatively close to each other (figure 14). Thus, the routing distance between two consecutive pick locations is relatively small in case products of only one segment are included in a given order. Analysis shows that a very large amount of orders consists of products that belong to only one segment (table 2). Naturally, the allocation of segments influences travel distance as well. Apparently, our method for allocating segments (section 5.4.3) performs well. Concluding, zoning in this case does not have such a negative impact on slotting.

Orders containing products of *n* (1…5) segments	**Percentage of total amount of orders**
Orders containing products of 1 segment	94.9%
Orders containing products of 2 segments	2.8%
Orders containing products of 3 segments	0.9%
Orders containing products of 4 segments	0.8%
Orders containing products of 5 segments	0.6%

Table 2: Per number of segments, the amount of orders (in %) is displayed.

6.6. Results of the direct link method

We apply the methodology described in section 5.7.2. Van Oudheusden *et al.* apply pair wise interchanges (i.e., swaps of product pairs) in step 3 of their heuristic until no change could be made resulting in a profit of more than 0.05% of the current value of the objective function (z). They apply their heuristic to only 1 aisle which contains a moderate amount of products. In our case, multiple aisles are used and the amount of products is significantly higher. Therefore, we use a threshold value of 1% instead of 0.05% in order to save CPU time. Van Oudheusden *et al.* (1988) observe that the most significant improvements were made during the first cycle of the heuristic, i.e., before calculating the $\widetilde{\mathbf{F}}_{ij}$ matrix for the second time (step 7 and 8 of the heuristic). Thus, it does not make sense to calculate the direct link matrix very often. We restrict the amount of recalculations to 3. For every cycle of the heuristic, we perform 1,000,000 random product swaps. We start from a storage assignment obtained with the COI policy (the results of the COI-policy are quite good). The heuristic is executed several times and the average results (for part 1 and 2 together) are provided in table 3.

Results SOP	Average distance (km)	Reduction (%)	CPU Time (hh:mm:ss)
Current distance	5292	-	00:00:04
Direct link method without zoning	3722	29.7%	02:42:00
Direct link method with zoning	3850	27.2%	02:50:00

Results after batching and splitting	Average distance (km)	Reduction (%)
Current distance	2029	-
Direct link method without zoning	1798	11.4%
Direct link method with zoning	1842	9.2%

Table 3: Results of the direct link method.

The heuristic achieves better results than the COI slotting method but at the cost of far more computation time (several hours). Again, the negative impact of zoning is small. The heuristic does not determine new picking tours after each swap. However, after a swap of two products the accompanying picking tours do change. Thus, an error occurs. Because many improvements are found in step 3, the size of this error grows (a large amount of direct link frequencies are no longer correct) and this probably explains the moderate results. Concluding, this heuristic does not perform well in a multi-aisle situation.

6.7. Results of order oriented product swapping

The previous method tries to deal with the order structure by determining direct link frequencies between two products. This heuristic considers complete orders, i.e., no surrogate measures are used. Two randomly chosen products are swapped from their location. Distances of order routes which contain exactly one of the two products are determined before and after the swap (5.7.3). Determining order routes is described in section 6.2. We use simulated annealing in combination with the 2-exchange process. Details can be found in appendix IV. The solution quality of this heuristic is very good. Although picking routes are determined very efficiently, computation times are still excessive. Therefore, the heuristic is executed only a couple of times and average results are shown in table 4.

Results SOP	Average distance (km)	Reduction (%)	CPU time (hh:mm:ss)
Current distance	5292	-	00:00:04
Order oriented product swapping without zoning	3186	39.8%	228:50:00
Order oriented product swapping with zoning	3265	38.3%	230:18:00

Results after batching and splitting	Average distance (km)	Reduction (%)
Current distance	2029	-
Order oriented product swapping without zoning	1637	19.3%
Order oriented product swapping with zoning	1665	17.9%

Table 4: Results of order oriented product swapping.

Slotting is usually not a time critical activity and the required computation time may therefore still be acceptable. However, if we use an order profile containing order data of several years, (and millions of order lines) computation times will become too high. Below, we discuss two heuristics that have this disadvantage to a far less extent.

6.8. Results of the interaction frequency heuristic

This heuristic requires 3 matrices. The distance matrix **D** and the order frequency matrix $\mathbf{F_{io}}$ are already discussed. The interaction frequency matrix $\mathbf{F_{ij}}$ stores all values of f_{ij} (i.e., how often appear product i and j on one order). It is constructed as follows. First, all orders are sorted on increasing product index number i. Then, per order two nested loops are used to calculate the interaction frequencies. The amount of non-negative interaction frequencies rapidly increases if the amount of order lines in the given order profile increases. We use an order profile with approximately 275,000 order lines. Consequently the amount of non-negative f_{ij} 's amounts to several millions. However, constructing the $\mathbf{F_{ij}}$ matrix does not take more than 30 seconds using Visual Basic for Applications. This is mainly due to the fact that the average amount of order lines per order (≈ 7) is small. After construction of the matrices, the methodology described in section 5.7.3.2 is applied. In step 4 of this heuristic, the non-negative interaction frequencies have to be sorted in decreasing order. Therefore, we use a Radix sorting algorithm in order to quickly sort the f_{ij} 's. This sorting method is very efficient for large lists of often identical integer values. Step 5b of the heuristic requires a value β that depends on the layout of the picking area. Experiments show that a value between 0.01 and 0.1 provides good results. We use a value of 0.08 for part 1 of the manual picking area and a value of 0.05 for part 2. Results are shown in table 5.

Results SOP	distance (km)	Reduction (%)	CPU time (hh:mm:ss)
Current distance	5292	-	00:00:04
Interaction frequency heuristic without zoning	3321	37.2%	00:01:56
Interaction frequency heuristic with zoning	3675	30.6%	00:02:10

Results after batching and splitting	distance (km)	Reduction (%)
Current distance	2029	-
Interaction frequency heuristic without zoning	1747	13.9%
Interaction frequency heuristic with zoning	1802	11.2%

Table 5: Results of the interaction frequency heuristic.

The results are much better than the COI and direct link method (especially in the case of single order picking) and the required CPU time is very modest. In case a larger order profile is used, only the construction of the $\mathbf{F_{ij}}$ matrix will take significantly more time. In that case, the required CPU time will still be considerably less than in the case of order oriented product swapping. We feel that the interaction frequency heuristic is suitable for many warehouse layouts. Naturally, the value of β needs to be changed accordingly.

6.9. Results of the interaction frequency based quadratic assignment heuristic

We use the same distance matrix **D** as in the previous sections to record all distances between two picking locations (the depot included). The order frequency matrix $\mathbf{F_{i0}}$ and interaction frequency matrix $\mathbf{F_{ij}}$ and can be used as well. After these matrices have been initialized, the heuristic starts by computing the first and second term of the objective function based on the current storage layout. For sake of clarity, we provide the objective function again:

$$\min_{a \in S} z(a) = \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} f_{ij} d_{ij}^{r}(a) + \alpha \sum_{i=1}^{I} f_{i0} d_{i0}^{r}(a) \qquad (1)$$

We use our rule of thumb (section 5.7.3.3) to determine the initial value of α. A 2-exchange procedure is used to swap products from their location. Simulated annealing based on the cooling schedule of Connolly (1990) (cf. appendix V) is used to minimize the objective function. We use a *sequential* neighbourhood search. Thus, the potential product pair swaps are examined in the order (1,2),(1,3),…,(1,I), (2,3),…,(I-1,I), (1,2),… and so on. In part 1 of the picking area, there are 6 different compartment sizes (appendix II). The assigned compartment sizes of two swapped products must match. Thus, in this case we have to run the annealing process for every compartment size. However, the compartments F, S and W are located near the depot *and* close to each other (appendix I). In our opinion, the savings obtained by swapping these products do not outweigh the cost of moving them. Furthermore, we can slightly reduce computation time by omitting these compartments. Thus, we only consider the compartment sizes A, B and C for part 1 of the picking area. For part 2, we consider all 4 compartment sizes. Table 6 displays the number of compartment sizes currently in use (F, S and W are omitted). Naturally, the number of occupied compartments (i.e., the number of products) relates to the problem size. Note that free (unassigned) compartments are spread evenly over the picking area *prior to* the optimization and thus are *not* considered during the annealing process.

compartment size	**# assigned compartments**	**Neighbourhood size N =1/2(I-1)I**
A (part 1)	1705	1452660
B (part 1)	2832	4008696
C (part 1)	2497	3116256
K (part 2)	211	22155
L (part 2)	426	90525
M (part 2)	472	111156
N (part 2)	223	24753

Table 6: Number of assigned compartments and accompanying neighbourhood sizes.

During each run of the annealing process, Connolly (1990) performs 50N swaps. The neighbourhood sizes of our problem instances for the compartments A,B and C are very large. Thus, we decrease the amount of swaps to N for every run. After each swap, the value of α needs to be adjusted in order to obtain equal weights of both terms (section 5.7.3.3). Because this is rather time-consuming, we adjust this value after 1/5N swaps. Part 2 of the picking area is considerably smaller, thus we apply 3N swaps for the compartments K up to N. In this case, we calculate the value of α after N swaps. The three sub problems above for part 1 are not independent, because products assigned to different compartment sizes have non-negative interaction frequencies f_{ij} (i.e., they appear on one order). The same holds for the 4 compartment sizes of part 2 of the picking area. Our heuristic picks a compartment size randomly and starts the annealing process. After completion, another compartment size is randomly chosen and the annealing process starts again until all compartment sizes are concluded. Every run of the annealing process stops after a predetermined number of steps. The start and final temperature are determined by using the suggested formulas of Connolly (1990) (appendix V). Table 7 displays the initial values of the objective function (z_0) and the initial α values for part 1 and part 2 of the picking area (the two parts are dealt with separately). In addition, the value of the objective function after completion of the annealing process (z_f) and the accompanying reduction is displayed. Note that the total reduction is displayed (thus after considering *all* compartment sizes) and that z does *not* denote the actual distance traversed. An initial layout is created using the interaction frequency heuristic.

Results without zoning	z_0 (m)	α	z_f (m)	Reduction (%)
Part 1 of the manual picking area	$1225.\ 10^6$	81.9	$910.1.\ 10^6$	25.7%
Part 2 of the manual picking area	$8.325.\ 10^6$	3.09	$6.811.\ 10^6$	17.1%

Results with zoning	z_0 (m)	α	z_f (m)	Reduction (%)
Part 1 of the manual picking area	$1433.\ 10^6$	83.5	$1275.\ 10^6$	11.0%
Part 2 of the manual picking area	$7.4.\ 10^6$	5.07	$6.0.\ 10^6$	18.9%

Table 7: This table shows the initial values of the objective function (z_0) (based on the interaction frequency heuristic) and the α values (based on our rule of thumb) for both parts of the picking area. Final values (z_f) and the reductions are shown as well. Note that these values do not represent the actual traversed distances.

The annealing process obtains a significant reduced objective value. The table below provides the *actual* average distances after applying the optimization technique several times (for part 1 and 2 together).

Results SOP	Average distance (km)	Reduction (%)	CPU time (hh:mm:ss)
Current distance	5292	-	00:00:04
QAP without zoning	3259	38.4%	05:08:00
QAP with zoning	3334	36.9%	05:36:00

Results after batching and splitting	Average distance (km)	Reduction (%)
Current distance	2029	-
QAP without zoning	1651	18.6%
QAP with zoning	1697	16.4%

Table 8: Results of the interaction frequency based quadratic assignment heuristic (QAP).

The results approach those obtained by order oriented product swapping. However, the required CPU time is far less. In case the order profile increases, only the construction of the $\mathbf{F_{ij}}$ matrix takes more time. Note that these results are obtained by simply using the minimal distances between every product pair (i.e., d_{ij}^m). We feel that the results can even be improved further by incorporating *routing-specific distances* (section 5.7.3.3). Further research is needed to model those distances correctly.

We applied our suggested rule of thumb for determining the value of α:

$$\alpha = \frac{\sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^m(a)}{\sum_{i=1}^{I} f_{i0} d_{i0}^m(a)} \qquad (2)$$

This value is adjusted several times during execution of the algorithm. However, we do not change the ratio of the first and second term of the objective function (1). As a result, both terms keep the same relative "weight". It is interesting to see what the effect is of altering the ratio between the first and second term and as a result α. That is, what happens to the quality of the obtained solution if the second term (i.e., the term that forces frequently ordered products to be placed not too far from the depot) gets for example a higher relative weight than the first term (or vice versa)? Figure 24 provides results (for single order picking without zoning for both parts of the manual picking area). A value of $1.8\alpha_0$ means that instead of equal relative weight of both terms (α_0 denotes the ratio in case equal relative weight of both terms is applied) , the second term now weighs 1.8 times more.

This can be seen as follows (for a given storage assignment *a*):

$$\alpha = 1.8\alpha_0 = 1.8\frac{\sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^r(a)}{\sum_{i=1}^{I} f_{i0} d_{i0}^r(a)} \Rightarrow z(a) = \sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^r(a) + (1.8\frac{\sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^r(a)}{\sum_{i=1}^{I} f_{i0} d_{i0}^r(a)})\sum_{i=1}^{I} f_{i0} d_{i0}^r(a)$$

Thus, the value of the objective function *z(a)* equals:

$$z(a) = \sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^r(a) + 1.8\sum_{i=1}^{I}\sum_{j=i+1}^{I} f_{ij} d_{ij}^r(a)$$

As a result, the second term weighs 1.8 times more than the first term.

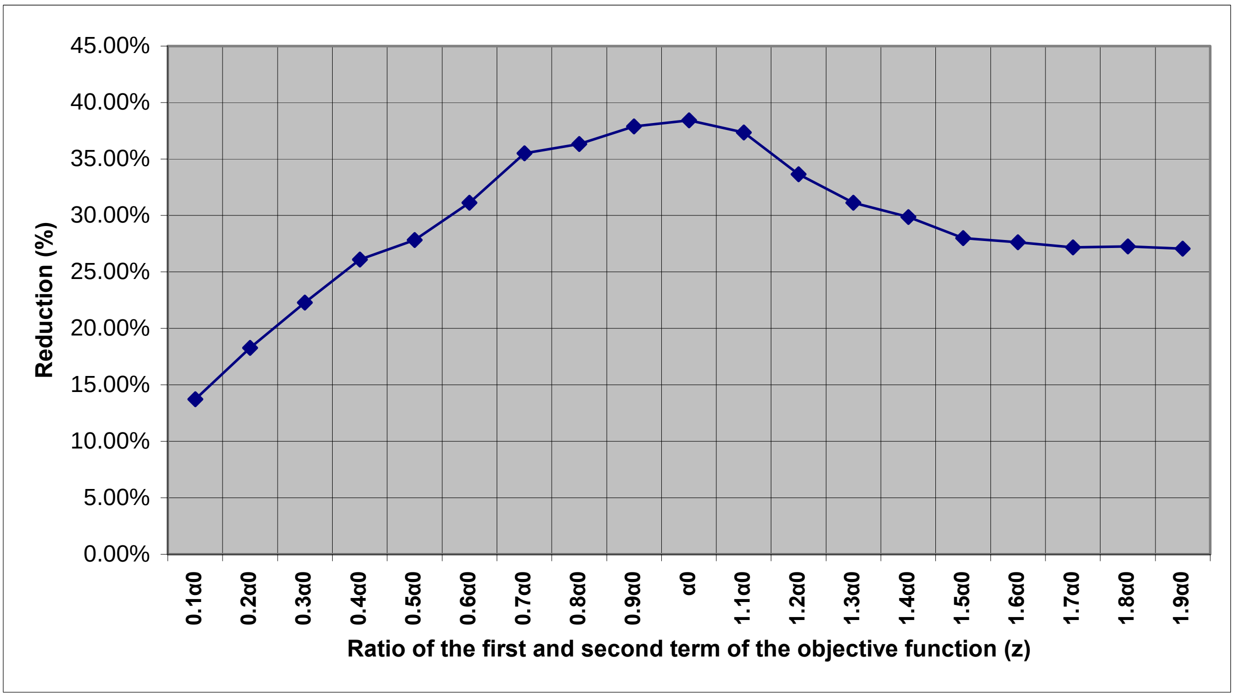


Figure 24: On the horizontal axis, different values of α are displayed. On the vertical axis, the distance reductions (in %) are displayed. Note that these reductions are with respect to the current traversed distance (for part 1 and 2 of the picking area together). In addition, single order picking without zoning is applied.

Apparently, small adjustments of the relative weights for both terms do not have a huge impact on the solution quality. As we increase the weight of the second term, the objective value approaches the solution obtained with a COI storage policy. An obvious reason for this observation is that placing frequently ordered products near the depot becomes far more important. A large relative weight of the first term deteriorates the solution quality. This can be explained as follows. Products which are often jointly requested are placed close to each other. However, frequently ordered products are more or less randomly placed in the warehouse. Naturally, this has a negative impact on solution quality. We conclude that our rule of thumb for determining α performs well.

6.10. Comparison of slotting methods

For sake of clarity, we summarize in figure 25 and figure 26 all results. Even the most simple slotting method (i.e., COI) achieves a significant reduction in travel distance. The direct link slotting method performs relatively moderate in our case because the warehouse consists of multiple aisles. This procedure is not well suited for multi-aisle situations. The order oriented product swapping method performs best, but the amount of CPU time is excessive. In addition, if we use more order data, the running time of this heuristic will increase even further. Interaction frequency based heuristics have this disadvantage to a far less extent. Indeed, only the construction of the $\mathbf{F}_{ij}$ matrix takes more time.

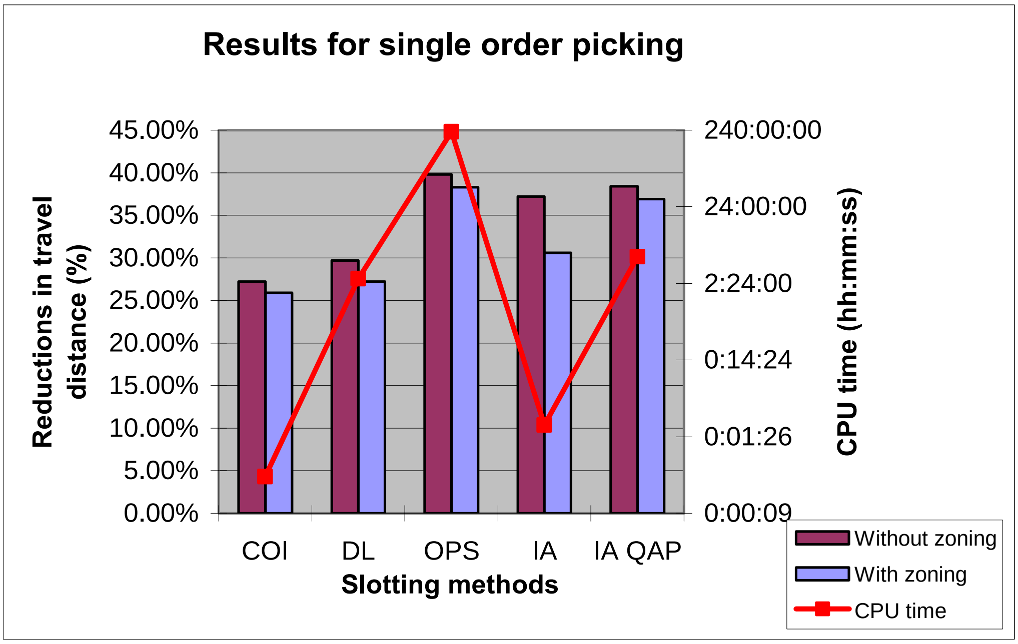


Figure 25: Results of all slotting methods in case of single order picking. (DL = direct link method, OPS = order oriented product swapping, IA = interaction frequency heuristic, IA QAP = interaction frequency based quadratic assignment heuristic)

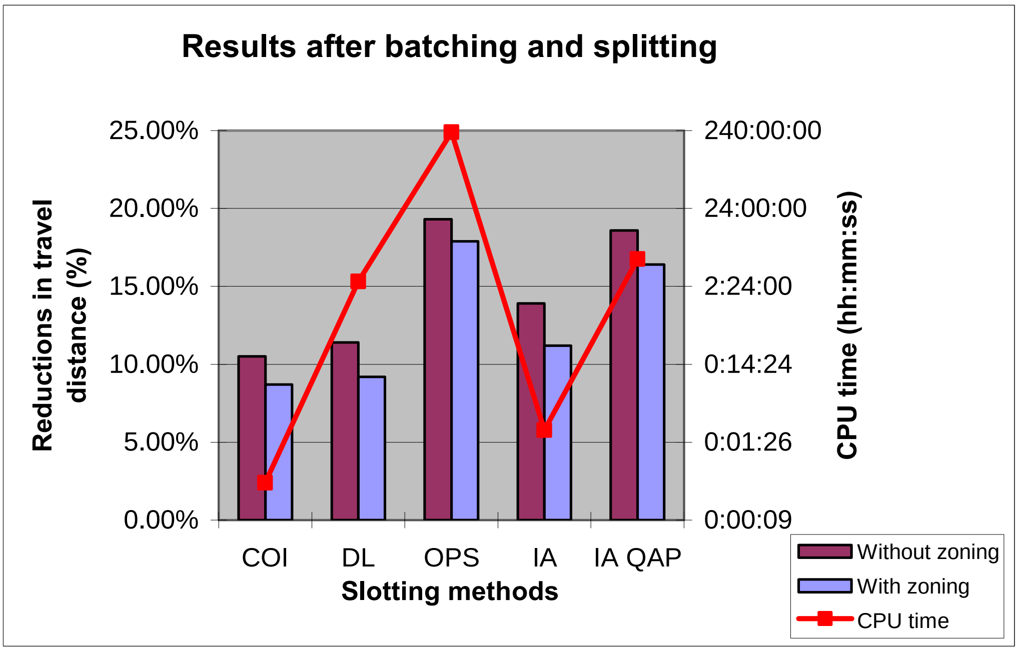


Figure 26: Results of all slotting methods after the batching and splitting process. (DL = direct link method, OPS = order oriented product swapping, IA = interaction frequency heuristic, IA QAP = interaction frequency based quadratic assignment heuristic)

The interaction frequency heuristic needs very little CPU time and achieves good results. The interaction frequency based quadratic assignment heuristic achieves very good results in a reasonable amount of CPU time. Although additional parameters are needed, our rules of thumb for determining those parameters perform well for the case at hand. As we already stated in the previous section, correctly determining *routing-specific-distances* between two storage locations probably improves the performance of the heuristic even further. We experienced in practice how difficult it is to determine those distances. Based on the amount of required computation time and the results, the interaction frequency based quadratic assignment heuristic performs best. Indeed, the amount of CPU time is not excessive (as opposed to order oriented product swapping) and results are very promising.

6.11. Conclusions

By using our slotting tool, the warehouse manager can analyze what reductions in travel distance are possible for a given order profile. Although we recommend to use the interaction frequency based quadratic assignment heuristic (IA QAP), the other methods can be used as well. The warehouse manager can adjust the restrictions concerning batching (i.e., capacity per pick cart in terms of weight and amount of order lines) and see what the effect is on travel distance. In addition, the slotting methods can be applied to the manual picking area with and without zoning.
Figure 26 clearly shows that all slotting methods obtain significant reductions in travel time even after batching and splitting of orders. Zoning does not have a huge impact on the performance of our slotting methods. This is due to the specific order structure (section 6.5).
The slotting tool also indicates the new locations of products. Naturally, reallocating products is a time-consuming process. An important question is if the profits of reallocating products outweigh the removal costs. This depends for a great deal on the “stability” of the order pattern. Management states that the order pattern is rather stable. Our slotting tool generates reallocation assignments of products (see figure 18). By incorporating those removal assignments in the WMS, order pickers could move products to their ‘optimal’ location during order picking. During the fall and winter seasons, the amount of orders is considerable less than in the spring and summer season. Order pickers can move the products to their new location in those periods. Because order pickers (excluding temporary workers) are employed on a regular basis, the costs of reallocating products in quiet periods (e.g. the fall season) are limited.

7 Conclusions and recommendations

In this chapter, the overall conclusions and recommendations are discussed. Suggestions for further research are also provided.

7.1. Conclusions

More than 350,000 orders are picked per year at the warehouse of Wolters-Noordhoff from three different areas: the bulk area, the dynamic picking area and the manual picking area. Order picking in the manual picking area is a costly process. Increasing the efficiency of this process can lead to considerable savings. A large part of order picking consists of traveling to pick locations. Because it does not add value, it is often a primary target for improvement. Wolters-Noordhoff wants to increase the efficiency of the order picking process by (re)allocating products in the manual picking area in such a way that travel distance is minimized. Our research focuses on this type of product allocation, i.e., slotting. Goal is to design a tool for efficiently slotting the manual picking area. The research was executed in the following way. First, we analyzed the current situation and identified which factors affect order picking efficiency. We learned that warehouse employees either batch orders (i.e., combining several small orders into one batch and collect the batch in one picking tour) or split large orders in sub-orders. Furthermore, order pickers use a specific routing policy to collect the requested items. Momentarily, no specific slotting strategy is applied.

Next, we performed literature research in order to see what slotting methods are available. Many papers in the field of warehousing focus on operational decisions like batching and routing. Indeed, combining several orders into one batch and collect them in one picking tour can lead to an increased efficiency of the order picking process. In addition of course, shorter picking routes lead to time savings as well. Literature on slotting is rather scarce. Heskett (1963) is the first researcher who deals with storage assignment. He introduced the Cube-per-Order-Index rule (COI). Basically, the idea is to store frequently ordered products close to the depot. This storage policy is optimal in case order pickers retrieve one product per picking tour (i.e., in case of a single-command order picking system). However, this is not true in case multiple products are collected in one picking tour (as is the case with Wolters-Noordhoff). Sometimes considerable savings are still possible using the COI-rule in a multi-command situation (e.g., Jarvis and Mcdowell (1991)). Intuitively, products that are often ordered together should be placed close to each other in the warehouse in case of a multi-command order picking system. Researchers like Frazelle (1989) and Rosenwein (1994) identify ‘closeness’ relationships between products by using cluster analysis. Amirhosseini and Sharp (1996) continued the work of Frazelle (1989). They introduced a generalized correlation measure that looks at the degree to which two or more products together fill customer orders. A shortcoming of these methods is that they do not explicitly work with travel distance. Instead, they use one of several surrogate measures of cluster strength. Furthermore, these researchers neglect the fact that there is a strong interrelationship between slotting and routing policies. Van Oudheusden *et al*. (1988) deal with these two problems simultaneously. Their slotting method looks at how often two products are picked before or after each other. They formulate the problem as a quadratic assignment problem (QAP) and use a local search method to minimize the QAP. Mantel, Schuur and Heragu (2007) introduce the concept of ‘Order Oriented Slotting’. They introduce slotting heuristics for which the way of allocating items to locations is directly related to the chosen routing strategy for picking the orders. The interaction frequency heuristic (IA) ranks interaction frequencies of product pairs (i.e., the frequency that a product pair occurs on orders). Products with a high interaction frequency should be placed close to each other (relative to the routing-policy-specific distance), but also in accordance with their order frequency. However, it does not become clear when a product is placed in accordance with his order frequency.

The interaction frequency quadratic assignment heuristic (IA QAP) tries to find a balance between placing product pairs with a high interaction frequency close to each other (relative to the routing-policy-specific distance) and placing frequently ordered products not too far from the depot. A

parameter is used to find an appropriate balance between those two objectives. Mantel, Schuur and Heragu (2007) that this balance parameter must be determined empirically.
In the next phase of our research, we adapted the slotting methods introduced by Heskett (1963), Van Oudheusden *et al*. (1988) and Mantel, Schuur and Heragu (2007) to the situation of Wolters-Noordhoff. We introduced a simple, but effective rule for the IA heuristic which clearly indicates when a product is placed in accordance with his order frequency. Furthermore, we introduced a rule of thumb for dynamically fine-tuning the balance parameter in the IA QAP heuristic.
In addition to these methods, we designed a new slotting heuristic for the OOS-problem: order oriented product swapping (OPS). Slotting is a very difficult combinatorial optimization problem. We tackled this problem by using simulated annealing in combination with a clever neighbourhood structure. As opposed to other slotting methods, OPS directly minimizes travel distance.
It should be noted that slotting is strongly interconnected with the batching and splitting problem. However, simultaneously considering batching, splitting and slotting is not very realistic and therefore we applied batching and splitting *after* the slotting process (i.e., we first assume that every order is processed separately). Order pickers at Wolters-Noordhoff use their own experience to batch or split orders. Only a few 'hard' restrictions must be taken into account. We made various assumptions for the batching and splitting process that reflect the current situation as well as possible. Batching diminishes the effect of slotting. This is not necessarily true for order splitting.
Management is also interested in the effect of placing products belonging to the same product group close to each other (i.e., zoning). Zoning can have a negative influence on the impact of slotting. The slotting tool also takes zoning into account.
In the last phase of this research, we constructed the slotting tool. We incorporated the different slotting methods in our tool. A sufficiently large order profile is used to determine the efficiency of the slotting methods. Results show that even the most simple slotting strategy (i.e., COI) obtains a significant reduction in travel distance. In addition, we see that zoning does not have a huge impact on the amount of travel reduction. This is possibly due to the order structure. The method introduced by Van Oudheusden *et al.* performs rather disappointingly. The fact that the warehouse of Wolters-Noordhoff consists of multiple aisles possibly hurts the efficiency of the heuristic (Van Oudheusden *et al.* apply their method to a single aisle). The order oriented slotting strategies perform very well. The current travel distance can be reduced by more than 15% both in cases with and without zoning applied. Our order oriented slotting method obtains the highest reduction in travel distance but at the cost of a large amount of computation time. The slotting methods introduced by Mantel, Schuur and Heragu and adapted to our situation are computationally less intensive. They almost obtain the same reduction in travel distance. In addition, their computation time does not depend for a great deal on the size of the order profile as opposed to our own slotting heuristic. Therefore, the interaction frequency based quadratic assignment heuristic is the most appropriate slotting method for Wolters-Noordhoff.
The warehouse manager can use the slotting tool to determine the traversed distance for a given order profile. He can insert order and product data, alter the batching and splitting rules (i.e., change the capacity of pick carts in terms of weight or maximum amount of order lines per batch) and possibly apply zoning. The possible reduction in travel distance by using the suggested slotting method is displayed as well. The tool also indicates the new 'optimal' locations of products. In addition, reallocation assignments of products are created. By incorporating the reallocation assignments generated by our slotting tool in the WMS, order pickers can gradually move products to their new locations in quiet periods (e.g. the fall season).

7.2. Recommendations

Obviously, the current assignment of products to locations in the manual picking area is not very efficient. Based on the assumptions we made concerning batching and routing, considerable savings in travel distance are possible both in cases with and without zoning. In addition, management states that the order pattern is stable. Therefore, we advice to implement the suggested slotting method, i.e., the interaction frequency based quadratic assignment heuristic. Management prefers the application of zoning (see figure 14). A reason amongst others is that order pickers are spread out more evenly over

the manual picking area. Since the negative impact of zoning on the amount of distance reduction is minimal, we have no objections to the application of zoning.
Naturally, it does not make sense to reallocate products very often. Slotting is a tactical decision which should be executed once or twice a year. Once the products have been allocated to their optimal locations, the warehouse manager should check after a predetermined period (e.g. once per year) with the aid of our slotting tool if the profits (in terms of distance reduction) of again reallocating products are high enough.

7.3. Suggestions for further research

Our tool focuses on slotting in the manual picking area only. Order pickers also need to collect products from the dynamic picking area (figure 3). A dynamic picking location is created when the total demand of a product exceeds the capacity in the manual picking area for that product. The dynamic picking area is situated close to the depot. As a result, a frequently ordered product is placed close to the depot. This results in reduced order picking time. There is a strong interaction between the assigned compartment sizes of products in the manual picking area and the amount of dynamic picking locations. Indeed, if a frequently ordered product is placed in a small compartment (thus only a small quantity is stored), a dynamic picking location for that product will be often created. This issue could be modeled and the effect of 'optimally' assigning compartment sizes to products could be investigated.
The slotting problem is a very interesting but difficult combinatorial optimization problem. This is due to the fact that issues like routing policies effect the slotting problem. By using the *routing-specific-distance* concept (for the interaction frequency based heuristics), we have tried to incorporate this. However, we experienced in practice how difficult it is to correctly determine those distances and used the minimal distances between product pairs (taking into account the aisle structure) instead.
Results are still very promising, but we feel that additional research concerning routing-specific distances is needed and could result in an even bigger improvement. Formulating the slotting problem as a QAP provides promising results. The size of our QAP instances are considerably larger than problem instances reported in literature. The accompanying solution space is huge. Grouping products in clusters could therefore be performed prior to solving the QAP. This reduces the problem size. Placing clusters (especially large ones) in the warehouse lay-out however, is not straightforward, because many practical issues (the lay-out of the warehouse, shelves lay-out, compartment sizes) must be taken into account as well.
A final interesting issue is to incorporate the costs of reallocating products in the slotting problem.

8 Reflection

We encountered some difficulties as well as some remarkable aspects during our research. Below, we shortly discuss these issues.

Goal of this research was to develop an efficient slotting tool for the manual picking area of Wolters-Noordhoff. An important advantage of this research problem was that it was scientific *and* business oriented. Indeed, applying optimization techniques for (all kinds of) business processes is a very important topic in the master Industrial Engineering and Management. The unambiguous formulation of this research problem was another advantage. This facilitated discussions with the graduation committee.
During our literature research, we discovered that there is a rather large amount of papers dedicated to routing problems in warehouses. Remarkably, many researchers state that the savings (in terms of travel distance) obtained by using optimal routes are rather moderate. We also applied the S-shape and largest gap routing policies to the warehouse of Wolters-Noordhoff. The difference in travel distance between these routing policies and the one currently used was indeed not significant. Savings obtained by slotting can be much higher. To our surprise, the amount of literature dedicated to slotting is rather limited. Only a few researchers in this field acknowledge that slotting and routing are interwoven. This makes slotting a difficult combinatorial optimization problem. We used the concept of routing-specific-distances between product pairs (Mantel, Schuur and Heragu, 2007) to deal with this interrelationship. The concept seemed useful, but we learned how difficult it is to determine those distances in practice.
A last remark concerns our own technique for solving the slotting problem (OPS). We were surprised that this method obtained very good results for reasonably large order profiles in an acceptable amount of time.
Concluding, this research project was a very meaningful experience. Putting (warehousing) theory into practice is a very challenging task. Literature on slotting is rather scarce and we hope that this thesis is a valuable contribution to the topic.

References

Aarts, E.H.L., Korst, J.H.M. (1989), *Simulated Annealing and Boltzmann Machines*, Chichester: Wiley

Amirhosseini, M.M., Sharp, G.P., (1996), "Simultaneous analysis of products and orders", *Manufacturing Science and Engineering*, MED 4, pp. 803-811

Ashayeri, J., Heuts, R., Tammel, B. (2005), "A modified simple heuristic for the p-median problem, with facilities design applications", *Robotics and computer-integrated manufacturing*, Vol. 21, pp. 451-464

Bartholdi, J.J., Hackman, S.T. (2006), "Warehouse and distribution science release 0.80", *Warehouse science website*, www.warehouse-science.com

Clarke, G. and Wright, W., "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, 1964, Vol. 12, pp. 568-581

Connolly, D.T. (1990), "An improved annealing scheme for the QAP", *European Journal of Operational Research*, Vol. 46, pp. 93-100

Frazelle, E.H., Sharp, G.P. (1989), "Correlated assignment strategy can improve order picking operation", *Industrial Engineering*, Vol. 4, pp. 33-37

Garfinkel, M. (2005), "Minimizing multi-zone orders in the correlated assignment problem, *Phd thesis*, Georgia Tech, Atlanta, USA

Hall, R.W. (1993), "Distance approximations for routing manual pickers in a warehouse", *IIE Transactions*, Vol. 25, No. 4, pp. 76-87

Heskett, J.L. (1963), "Cube-per-order index – a key to warehouse stock location", *Transport and distribution management*, Vol. 3, pp. 27-31

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

Ji, P., Liu, H., Wu, Y. (2006), "A solution method for the Quadratic Assignment Problem (QAP)", *ORSC & APORC* , pp. 106-117

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983), "Optimization by Simulated Annealing", *Science*, Vol 220, No. 4598, pp. 671-680

De Koster, M.B.M, Van der Poort, E.S., Wolters, M. (1999), "Efficient orderbatching methods in warehouses", *International Journal of Production Research*, Vol. 37, No. 7, pp. 1479-1504

De Koster, M.B.M., Le-Duc, T. (2005), "Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse", *International Journal of Production Reserarch*, Vol. 43, No. 17, pp. 3561 – 3581

De Koster, R., Le-Duc, T., Roodbergen, K.J. (2006), "Design and control of warehouse order picking: a literature review", *ERIM Report series research in management*

Krishnakumar, B., Malmborg, C.J. (1989), "Optimal Storage Assignment Policies for Multiaddress Warehousing Systems", *IEEE Transactions on systems, man and cybernetics*, Vol. 19, No. 1, pp. 197-204

Liu, C.M. (1999), "Clustering techniques for stock location and order-picking in a distribution center", *Computers and Operations Research*, Vol. 26, pp. 989-1002

Liu, C. (2004), "Optimal storage layout and order picking for warehousing", *International Journal of Operations Research*, Vol. 1, No. 1, pp. 37-46

Mantel, R.J., and Rouwenhorst, B. (1998), "A Warehouse Design Model", *Progress in Material Handling Research: 1998*, R.J. Graves *et al.* (eds), Material Handling Institute, Charlotte, NC, pp. 315-324

Mantel, R.J., Heragu, S.S. and Schuur, P.C. (2007), "Order Oriented Slotting: A new assignment strategy for warehouses", *European Journal of Industrial Engineering*, Vol. 1, No. 3, pp. 301-316

Van Oudheusden, D.L., Tzen Y.J, Ko, H. (1988), "Improving storage and order picking in a person-on-board AS/R system: A case study", *Engineering Costs and Production Economics*, Vol. 13, pp. 273-283

Pan, J.C., et al., (2005), "Storage assignment in a narrow-aisle warehouse with multi-picker situation", *Journal of industrial and business management*, Vol. 1, No. 1, pp 1-10

Pardalos, M., Resende, M.G.C., "A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 1-21

Petersen, C.G. (1997), "An evaluation of order picking routing policies", *International Journal of Operations & Production Management*, Vol. 17, No. 11, pp. 1098-1111

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

Petersen, C.G., Aase G. (2003), "A comparison of picking, storage and routing policies in manual order picking", *International Journal of Production Economics*

Ratliff, H.D., and Rosenthal, A.S. (1983), "Orderpicking in a rectangular warehouse: a solvable case of the traveling salesman problem, *Operations Research*, Vol. 31, No. 3, pp. 507-521

Roodbergen, K.J., De Koster, R. (1998), "Routing order pickers in a warehouse with multiple cross aisles", *International Journal of Production Research*, Vol. 39, No. 9, pp. 1865-1883

Rosenwein, M.B. (1994), "An application of cluster analysis to the problem of locating items within a warehouse", *IEEE Transactions*, Vol. 26, pp. 101-103

Rouwenhorst, B. et al. (2000), "Warehouse design and control: Framework and literature review", *European Journal of Operational Research,* Vol. 122, pp. 515 - 533

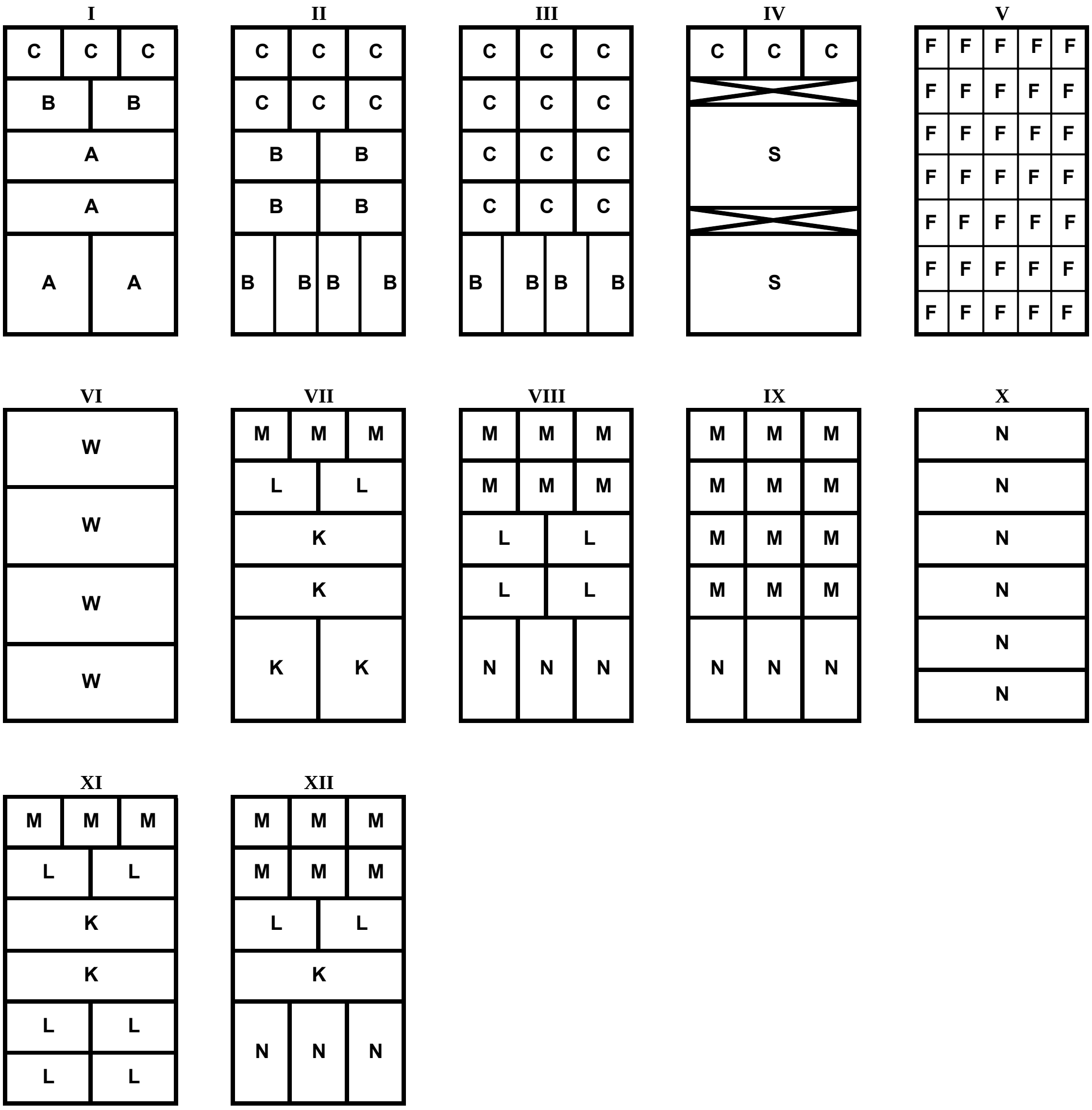
Taillard, E. (1991), "Robust taboo search for the quadratic assignment problem", *Parallel Computing*, Vol. 17, pp. 443-455

Tompkins, J.A., et al. (2003), *Facilities Planning*, New York: John Wiley & Sons

Wäscher, G. (2004), “Order picking: a survey of planning problems and methods”, *Supply chain management and reverse logistics*, pp. 323–347

Appendix I Overview manual picking area

Appendix II Shelves lay-out and compartment sizes



The *front views* of all shelves in the manual picking area are displayed above. Shelves I – VI (w *x* h *x* d = 1.3m *x* 2.4m *x* 0.6m) are located in part 1 and VII-XII (w x d x h = 1m x 2.4m x 0.6m) in part 2 of the picking area. The capital letters refer to the compartment sizes. The table below gives an overview.

Compartment code	Size (as part of the total volume of a shelf)
A (part 1)	1/6
B (part 1)	1/12
C (part 1)	1/8
F (part 1)	1/35
S (part 1)	1/4
W (part 1)	1/3
K (part 2)	1/6
L (part 2)	1/12
M (part 2)	1/18
N (part 2)	1/9

Appendix III OOS for S-shape routing policy

Consider the single block warehouse depicted below:

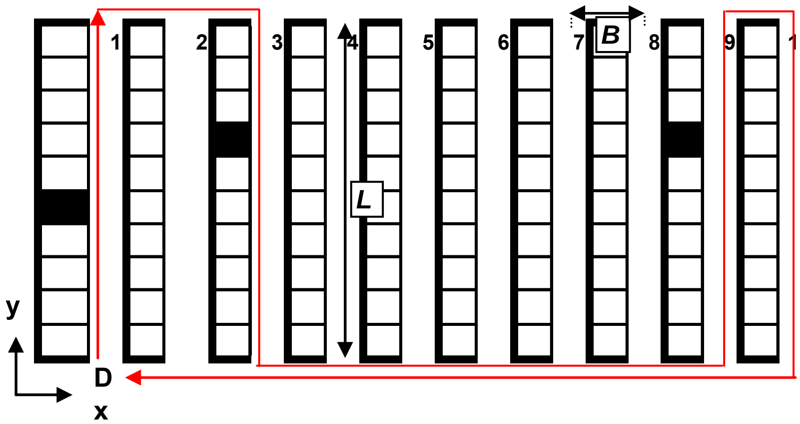


Figure 27: S-shaped routing.

The depot is situated bottom left and a S-shape routing policy is applied. Thus, every aisle that contains a pick locations is traversed entirely. In case of an odd amount of pick aisles, an additional aisle is idly traversed. Products can only be picked from the right side of an aisle. The order oriented slotting (OOS) problem is as follows. Given are a set of products i = 1,2,..I. and a set of pick orders k = 1,2,…K. The order set O_k consists of the products located in the warehouse (i.e., $O_k \subset \{1,2,...,I\}$). The objective is to locate products in such a way that travel distance is minimized. We first introduce the required parameters and variables.

Parameters:

Aisles $g = 0,1,..G$
C_g = number of storage points of aisle g

Variables:

U_k = rightmost aisle where a product of O_k is located
V_k = half of the number of aisles traversed in picking order k

$$X_{ig} = \begin{cases} 1 \text{ if product i is located in aisle g} \\ 0 \text{ otherwise} \end{cases}$$

$$Z_{kg} = \begin{cases} 1 \text{ if order k has a product in aisle g} \\ 0 \text{ otherwise} \end{cases}$$

An ILP can now be formulated for a single block warehouse with S-shape routing policy.

$$\min \sum_{k=1}^{K} (2BU_k + 2LV_k) \qquad (1)$$

$$s.t.$$

$$\sum_{i \in O_k} X_{ig} \le |O_k| . Z_{kg} \qquad \forall g, k \qquad (2)$$

$$\sum_{g=0}^{G} Z_{kg} \le 2V_k \qquad \forall \text{k} \qquad (3)$$

$$\text{g}.\text{Z}_{\text{kg}} \le U_k \qquad \forall k \forall \text{g} \ge 1 \qquad (4)$$

$$\sum_{g=0}^{G} X_{ig} = 1 \qquad \forall i \qquad (5)$$

$$\sum_{\text{i}=1}^{I} X_{ig} \le C_{\text{g}} \qquad \forall g \qquad (6)$$

$$\text{X}_{\text{ig}}, Z_{kg} \in \{0,1\} \qquad \forall i, k, g \qquad (7)$$

$$\text{U}_{\text{k}}, V_k \text{ integer and nonnegative } \quad \forall k \qquad (8)$$

The objective function (1) calculates the distance in x –and y-direction for all orders. Restriction (2) connects order and product variables. Constraint (3) ensures that the number of aisles traversed is even. Constraint (4) determines the farthest picking aisle and constraint (5) ensures that each product is assigned to exactly one aisle. Restriction (6) ensures that the capacity of each aisle is not violated.

The ILP formulation above can only be solved for small problem sizes. For large instances, heuristics are needed. In chapter 6, we applied the interaction frequency based quadratic assignment heuristic:

$$z = \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} f_{ij} d_{ij} + \alpha \sum_{i=1}^{I} f_{i0} d_{i0} \qquad (9)$$

The *routing-specific-distances* in the case of an S-shape routing policy can be defined as:

$$d_{i0} = \text{aisle to which product } i \text{ is allocated.}$$

$$d_{ij} = |d_{i0} - d_{j0}| \text{ (i.e. the number of aisles between product } i \text{ and } j.$$

Of course, (9) does not represent the actual travel distance because d_{ij} only denotes the difference in aisle numbers. In addition, distances between *all* product pairs are taken into account. However, a reduction of z (9) should also result in a reduction of the actual travel distance. Order and interaction frequencies (i.e., how often appear product i and j on one order) are represented by f_{i0} and f_{ij}. The value of α (i.e., the ratio of the two terms) is initialized as follows:

$$\alpha = \frac{\sum_{i=1}^{I} \sum_{j=i+1}^{I} f_{ij} d_{ij}}{\sum_{i=1}^{I} f_{i0} d_{i0}} \qquad (10)$$

We start from the warehousing situation above (aisle width: 1m, aisle length: 10 *x* 1m) and create order data consisting of 3000 orders with a maximum order size of 15 order lines. Maximum order frequency amounts to 100. Simulated Annealing (with the cooling schedule of Connolly (1990), is used to minimize z (a solution based on the COI rule is used as a start solution). The value α is adjusted after a certain amount of iterations. We executed each method 50 times. Results are provided below.

Random allocation		**COI Allocation**			**Interaction Frequency QAP Allocation**		
Average (km)	Standard Deviation	Average (km)	Standard Deviation	Reduction w.r.t. random	Average	Standard Deviation	Reduction w.r.t. COI
134,2	3,5	115,6	3,4	9,7%	112	3,4	6,8%

The interaction frequency based quadratic assignment heuristic performs well in this case obtaining an average reduction of 6.8% with respect to the in practice frequently used COI based allocation.

Appendix IV Simulated annealing

The concept of simulated annealing is based on the physical annealing process of solids. It is an combinatorial optimization technique in which improvements (in this case reductions in travel distance) are always accepted, while deteriorations are also accepted to a limited extent.
Consider the following notation:

S = finite set of solutions, i.e., in this case the set of storage assignments.
f = total travel distance.
i = current solution.
j = new solution.
c_0 = start temperature / control parameter.
c_k = temperature at the k^{th} iteration.
L = number of iterations at each temperature (also called the Markov chain).
L_k = number of iterations generated at the k^{th} iteration.

Here, the objective is to minimize travel distance by selecting the best alternative of the finite set of solutions, thus: $\min_{i \in S} f(i)$. An *acceptance criterion* is used to determine whether j is accepted from i by using the following *acceptance probability* :

$$\mathrm{P}_c\{accept\ j\} = \begin{Bmatrix} 1 & if\ f(j) \leq f(i) \\ \exp(\frac{f(i) - f(j)}{c}) & if\ f(j) > f(i) \end{Bmatrix}$$

During the execution of the heuristic, the control parameter c decreases. Consequently, the probability of accepting a deteriorated solution also decreases.
The procedure can be summarized in pseudo code as follows (Aarts & Korst, 1989):

```
Procedure Simulated Annealing

Begin
      INITIALIZE (i_start, c_0, L_0)
      k := 0
      i := i_start
      repeat
            for l := 1 to L_k do
            begin
                  GENERATE (j from S_i)
                  if f(j) ≤ f(i) then i:=j
                  else
                  if exp ((f(i) - f(j)) / c_k) > random[0,1] then i:=j
            end
            k := k + 1
            CALCULATE_CONTROL (c_k)
      until stopcriterion
end
```

An important question remains what values must be taken for c_0 and L. After each iteration, the control parameter c has to decrease. Thus, an appropriate decrease factor α has to be determined as well. Furthermore, we need a stop criterion in order to terminate the heuristic. This set of parameters is called a *cooling schedule*. The search for adequate cooling schedules has been the subject of study in many papers. Every combinatorial optimization problem requires a different cooling schedule. We use

a cooling schedule based on certain simple empirical rules. It is proposed by Kirkpatrick, Gelatt and Vecchi (1982,1983).
The decrease factor α is often modeled as a constant smaller than but close to 1. Typical values lie between 0.8 and 0.99. We have chosen a value of 0.94.
The initial value of the control parameter (c_0) should be chosen such that virtually all transitions (swaps) are accepted. This can be accomplished by requiring that the *initial acceptance ratio* $\chi(c_0)$ is close to 1. The acceptance ratio indicates the ratio between accepted swaps and the total amount of proposed swaps. Because swaps that result in an improvement are always accepted, we need to look at the number of accepted swaps (*N*) that deteriorate *f*. The total cost of deterioration equals Δf. The average cost per accepted deterioration equals $\Delta f / N$. The initial value of the control parameter can now be determined as follows:

$$c_0 = -(\frac{\Delta f}{N})\ln(\alpha) \qquad (1)$$

We perform a small fraction of the total number of iterations in order to determine the initial value of the control parameter. The length of the Markov chain must be chosen such that all neighbourhood solutions can be reached. Because the amount of neighbourhood solutions is rather large, we choose a length of 10,000 iterations. The heuristic terminates if one of the following situations occur:

1. The acceptance ratio $\chi(c_k)$ is equal to or less than 0.05.
2. The current temperature equals 1.
3. The best solution found does not change after 5 consecutive Markov chains.

If we choose the initial value of the control parameter in such a way that all swaps are accepted, the quality of the starting solution does not matter much. This means that we can start with a random solution. However, experiments show that simulated annealing in this case requires a large amount of computation time. Furthermore, we observe that the solution space is huge (8,800 products). The heuristic can escape rather easily from local minima. We therefore decide to set the initial acceptance ratio considerably lower than 1.The quality of the starting solution now has a greater impact. Section 6.5 shows that slotting based on the COI-policy obtains good results and therefore we start with that solution.

Appendix V Solution methods for the QAP

The quadratic assignment problem (QAP) was introduced by Koopman and Beckmann in 1957 as a mathematical model for the location of indivisible economical activities. QAP is often used to describe a location problem. Let us assign n facilities to n locations with the cost being proportional to the flow between the facilities multiplied with their distances. The objective is to allocate each facility at a location such that the total cost is minimized. Thus we are given two n x n matrices, the flow matrix **F** (containing all values of f_{ij}) and a distance matrix **D** (containing all values of d_{ij}). The QAP can now be written as follows:

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{i=1}^{n} f_{ij} d_{\pi(i)\pi(j)} \qquad (1)$$

Here, S_n is the set of permutations of $\{1,2,...,n\}$. Each individual product $f_{ij} d_{\pi(i)\pi(j)}$ is the cost caused by assigning facility $\pi(i)$ to location i and facility $\pi(j)$ to location j. A slightly different problem also addressed as a QAP is the following. Besides the two matrices **F** and **D** we are given a third matrix **C** (containing all values of c_{ij}), whose c_{ij} is the cost of placing facility i at location j. The problem is now as follows:

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{i=1}^{n} f_{ij} d_{\pi(i)\pi(j)} + \sum_{i=1}^{n} c_{\pi(i)i} \qquad (2)$$

Many researchers devote their attention to solution methods for the quadratic assignment problem. This type of optimization problem is known to be NP-hard. Only small problem sizes ($N = 50$) can be solved to optimality. Thus, many heuristics have been introduced for larger problems. The most successful ones are based on the following combinatorial optimization techniques:

1. Tabu search.
2. Greedy randomized adaptive search procedure (GRASP).
3. Simulated annealing.

Tabu search is a local search method, which is able to escape from local minima. Based on an initial solution, all the neighbour solutions are evaluated. The best of them is accepted. This does not have to be a better solution. To prevent cyclical exchanges, the last k exchanges are stored in a tabu-list. These exchanges are not allowed ('tabu'). Every time a new exchange is added to the front of the list, the exchange at the end of the list is deleted. The algorithm stops when improvements are not realized after a predetermined number of exchanges *or* when all the (direct) neighbours are in the tabu list. Often, an *aspiration level* is used. This is a rule, that allows an exchange even though it is listed in the tabu list. For example, if a 'forbidden' exchange results in an improvement of the best solution so far, it is accepted. The best known tabu search algorithm for a QAP is the robust tabu search algorithm of Taillard (1991). This algorithm is based on a 2-opt best-improvement local search technique. It uses a variable tabu list size. Good results are obtained with problem sizes up to $N = 64$.

GRASP is a heuristic and possesses four basic components: a greedy function, an adaptive search strategy, a probabilistic selection procedure and a local search technique. Pardalos and Resende (1994) apply this heuristic to 88 QAP instances and report good solutions.

Simulated annealing is successfully applied to several QAP instances by Connolly (1990). A 2-exchange procedure is used to swap units. He provides two reasons why a sequential search of the neighbourhood instead of a random search (many 2-exchange procedures are based on random swapping) could provide better results. Firstly, potential improvements might be missed at low temperatures because of the random nature of the search. Secondly, attempts to move away from local optima could be obstructed by the premature repeal of uphill escape attempts (Connolly, 1990, p94). Numerical experiments subscribe this theory. Connolly provides an optimized cooling schedule based on several rules of thumb. The start temperature T_0 (control parameter) is chosen such that: $T_0 = \delta_{\min} + \frac{1}{10}(\delta_{\max} - \delta_{\min})$. The value $\delta_{\min}$ denotes the minimum difference of the objective value before and after a swap and $\delta_{\max}$ denotes the maximum difference (both differences > 0, because improvements are always accepted). The author argues that this way the start temperature is not 'too hot' (i.e., all bad uphill moves are accepted with the result that a good solution cannot be obtained within a reasonably amount of time) and not 'too cold' (i.e., the heuristic quickly reaches a local optimum and cannot escape from it). Thus, the initial acceptance ratio is *not* set equal to 1 as opposed to many other cooling schemes found in literature. The final temperature T_f is set equal to $\delta_{\min}$. A number of $\frac{1}{100}M$ *random* swaps are executed to determine T_0. M denotes the *total* number of swaps and is set equal to 50K. K denotes the size of the neighbourhood. Because a sequential search is used, it is equal to ½(N(N -1)) (N = number of units). All swaps improving the current objective (i.e., $\delta \leq 0$) are accepted while uphill steps of size $\delta > 0$ are accepted with probability $e^{-\delta/T}$ by drawing a random number X from a uniform [0, 1] distribution and accepting the swap if X ≤ $e^{-\delta/T}$. The temperature is controlled by a parameter β via the following recurrence relation: $T_{i+1} = T_i/(1+\beta T_i)$. Because the author wants to complete the algorithm in M (=50K) steps, β equals: $(T_0 - T_f)/(MT_0T_f)$. Note that after *each* swap, the temperature is decreased as opposed to many other cooling schedules where the temperature decreases after a certain amount of iterations (i.e., the length of the Markov chain). Numerical experiments (with problem sizes up to N = 100) show that the cooling schedule of Connolly (1990) performs better (solution quality and reduced CPU time) than other cooling schedules applied to quadratic assignment problems.

The optimization techniques described above (which are often used) seem promising. However, the size of our problems instances are significantly larger than problem sizes reported in literature. Therefore, we question how the solution methods described above perform in our case. The warehouse manager (user of the slotting tool) has no experience with combinatorial optimization. Thus, a heuristic which requires no additional parameter setting would be ideal. To our knowledge, there exist no suitable combinatorial optimization technique (with the exception of a simple local search method) that does not require additional parameters. The rules of thumb for a cooling schedule, introduced by Connolly (1990) however, can be automated and therefore require almost no additional input of the user (i.e., setting the cooling schedule parameters). In addition, this method obtains good results for a large variety of QAP instances. Therefore, we decide to use this method for our QAP formulation of slotting.