Warehouse Operations Revisited - Novel Challenges and Methods

Susanne Wruck

Reading committee:

prof.dr. W. Dullaert
prof.dr. Jan C. Fransoo
prof.dr. Kai Furmans
prof.dr. M.B.M. De Koster
prof.dr. Leen Stougie

VRIJE UNIVERSITEIT

Warehouse Operations Revisited
-
Novel Challenges and Methods

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Economische Wetenschappen en Bedrijfskunde
op maandag 30 juni 2014 om 11.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door
Susanne Wruck
geboren te Berlijn, Duitsland

promotoren: prof.dr. I.F.A. Vis
prof.dr. J. Boter

Contents

Chapter 1

Introduction

This chapter will review the development of e-commerce in the past years and shed light on its particular characteristics in comparison to in-store buying. Given the increase of sales via online channels for many vendors, we explain and discuss the need for adaptation in the logistics operations to facilitate efficient and service-oriented order fulfillment. The chapter will introduce the major roles of commercial warehouses in supply chains and pinpoint the novelties and challenges for warehouse operations being accompanied by Internet purchases. It will further introduce the case of a library warehouse, which is used for numerical experiments in the thesis, and will explain its distinctions but also its commonalities with e-commerce warehouses that make a library warehouse a suitable example for our considerations. Lastly, this chapter will clarify the purpose of this dissertation and indicate its research contributions and practical implications.

1.1 Today's e-commerce business

During the last two decades e-commerce has become an essential part of the retailing industry (Gefen, 2000; Ha and McGregor, 2013). For consumers the possibility to purchase a wide range of products via the Internet is widely accepted and common (Ofek et al., 2011; Doherty and Ellis-Chadwick, 2010). Besides the most obvious reason, namely the convenience to shop at home and the accompanied time savings, the reasons for the growing number of Internet purchases are manifold. Consumers are not restricted by opening hours. They can browse through websites and place their orders whenever it is comfortable for them. The products are usually delivered within very short delivery times and directly to the consumer's house. Companies permit several weeks - often even months - to return products free of charge. Online shopping

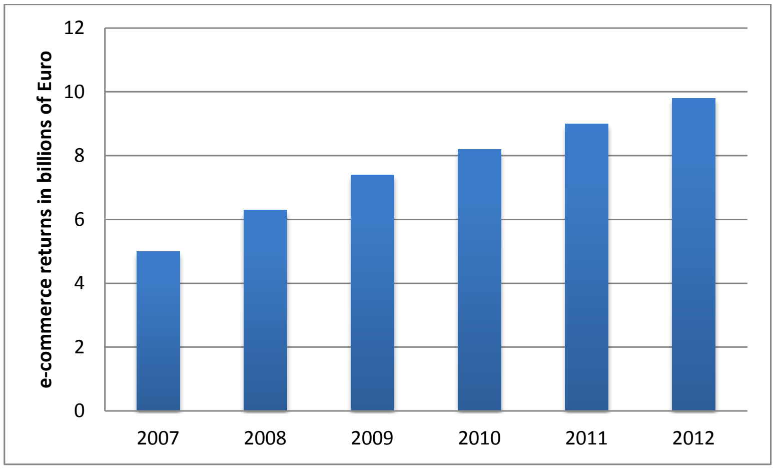


Figure 1.1: E-commerce returns in the Netherlands, Source: Internet Logistiek: De Sleutel tot verdere e-commerce groei, jaarverslag 2012

is very easy regarding the comparison of products (Keeney, 1999), discount offers, and prices of different vendors. Typically even a larger product variety is offered than in in-store assortments, because larger product collections are possible if the products are stored in warehouses rather than being presented in retail stores; and a larger assortment has been found to attract customers (Brynjolfsson et al., 2003). Moreover, services of online retailing, such as recommendations of products and a variety of payment opportunities make Internet shopping very attractive for today's consumers.

High speed Internet connections in private households have facilitated this strong growth of e-commerce, which, therefore, has a relatively short history. For example, Figure 1.1 shows the development of the e-commerce growth in sales in The Netherlands since 2007 showing an increase of 96% within five years. This strong increase can also be observed in other countries.

Considering the logistics perspective, the increase of Internet sales has led to changes in the fulfillment of demand. Companies' supply chains have to adjust to the transition from processing large stock-keeping units (SKU) for in-store buying toward dealing with numerous small orders to be delivered to single end customers. Next to that, online retailing has raised the amount of product returns for which optimal processing channels still have to be designed. The underlying supply chain concepts differ considerably from traditional supply chains of in-store buying. Burt and Sparks (2003) list the major characteristics of e-retailing. Those include the opportunity to operate through multiple marketing channels, new forms of non-price competition (e.g., website quality and delivery times), and the central importance of customer loyalty. Logistics operations obviously become more complex given highly demanding customers. The storage of goods in external distribution centers might be employed to save time and costs. However, in this case time and cost restrictions are not only issues of single companies, but also there is an impact on their business partners being responsible for production, supply, and replenishment. Given the convenience for the customer to choose products from the collections of many providers at the time, competition is higher and the need to excellent service performance has become inevitable for maintaining a competitive position.

These circumstances motivate and require a reconsideration and potentially adjustments of the operational processes within supply chains of e-commerce businesses with respect to efficiency-based and service-oriented performance measurement. A central component of e-commerce supply chains are the warehouses in which products from various manufacturers and suppliers are stored, picked, consolidated, packed and bundled for shipment. This thesis will focus on aspects that have an impact on warehouse performance in retail supply chains. Before we discuss those in more details we illustrate the problem of numerous product returns being one of the major challenges that emerged in the course of e-commerce growth. Product returns will therefore be a central issue of this work.

1.2 Customer order returns

The convenience to shop online is usually accompanied with the opportunity for customers to send arbitrarily many products back to the warehouse within a certain time period after delivery. In most cases customers have the right to return products by law within a certain period, or when the product is damaged or mistakenly delivered. However, many vendors also use their return policies to create a competitive advantage by offering very flexible return policies such as returns guarantees long after the order, free of costs, and without the request for return reasons. Generous return polices have been shown to serve as incentives for customers to order products, since all undesired deliveries can always be sent back. Chang et al. (2013) recently found a strong positive effect of good return policies on consumer trust; already Ramanathan (2011) confirmed a strong effect of good return policies on customer loyalty.

In spite of the advantages that product return policies have for companies to attract customers, the actual product returns are problematic from a logistics perspective. Mostard and Teunter (2006) reported that return rates can be as high as 75% depending on the product category. These products have to be unpacked, inspected, repackaged or discarded, and reinserted in the warehouse stock before they can be resold (Su, 2009). Next to that, product returns cause additional inventory holding costs as long as they are not available for purchase again. If a product is not reincorporated in the warehouse stock yet, it can usually not be ordered by other customers, since the allowed time windows for order picking are very tight and those deadlines cannot be met if a product are not at its designated storage location. Quick and reliable inventory updates therefore should include the returned products, and thus assist in maintaining a good overview of the collection in stock which allows for smart replenishment policies, while avoiding obsolescence and stock-outs. Product returns are causing high operational costs (Shulman et al., 2010), so that the high return rates have become problematic for online vendors. Nevertheless, many researchers remark that product return issues have not been studied extensively in the past (Bonifield et al., 2010) and that the impact of returns is ignored or not well-understood yet (Mollenkopfa et al., 2011). Also, Bernon et al. (2013) suggest to improve the product return processing.

As indicated above, many of the problems accompanied with product returns occur in the logistics processing. With respect to shipment, companies have to develop low-cost policies to transport the products back to the warehouse. For the convenience of the customers the return shipment is often arranged by a third party and at the expense of the vendor, so that the customer can easily bring the package to a post office in the neighborhood or arrange that the package is picked up at the customer's house. When the returned product arrives at a warehouse it has to be processed. It might be prepared for reorder, repackaged, shipped to the manufacturer, inspected, or even discarded. Thereby the processing of returns often interferes with the remaining warehouse operations to handle the outgoing orders. Small numbers of returns might be handled with auxiliary cost and labor effort in order to maintain efficient and undisturbed operations for the outgoing flow of customer orders. Yet if the number of returns is very high, this option might become enormously expensive. Therefore, the first three research projects in this thesis deal with operational methods for warehouses that face a large number of product returns. We suggest methods to integrate returns in an efficient manner with the classical operations to fulfill customer orders whenever this is appropriate. The following section provides a more detailed explanation of the tasks and operations of and within a warehouse, which will reveal the opportunities to deal with product returns by combining it with customer orders.

1.3 Warehouses

Warehouses are a central component in e-commerce supply chains. Many webstores are selling products from different manufacturers and work with supplying business partners so that the assortment of offered products is brought together in a warehouse. The warehouse is responsible for inventory management and replenishment of orders, as well as storing, sorting, and potentially repackaging of goods. Most importantly, the warehouse is managing customer orders, which consists of picking the order, consolidation and sorting, packaging and labeling, as well as it is managing in the information throughout the supply chains (e.g. reporting stock-outs or order status information for the customer). We distinguish between external and internal tasks of warehouses, which are explained in more details in the following.

1.3.1 Warehouse tasks

Warehouses can adopt a variety of roles within a supply chain, depending on the type of products to be stored and the function of the other supply chain members. These roles might go beyond the most obvious task to temporarily store goods. Heragu (2008) classifies these roles as follows:

- *Put together customer orders:* This describes typical e-commerce situations in which the warehouse stores products in large quantities received from one or multiple suppliers. In response to customer orders these products are grouped and sent directly to the customer.
- *Serve as a customer service facility:* The warehouse is in charge for handling replacement or reparation of sold goods as well as it can provide after-sales services.
- *Protection of goods:* Naturally an important role of warehouses is to protect the stored products against external influences, such as theft, fire, or floods.
- *Segregate hazardous or contaminated materials:* It might be necessary to store some materials apart a manufacturing system. In such cases a warehouse can be used to store these materials.
- *Perform value-added services:* If warehouses receive products in large quantities (re-) packaging to small quantities might be required. Also testing and inspection of the received products can belong to a warehouse's tasks.
- *Inventory:* Inventory control, safety stock decisions, and replenishment policies are tasks that a warehouse is facing. Especially in e-commerce, the assortment of products rapidly changes and products are often in stock in small quantities to avoid obsolescence of the product collection. Accurate inventory management are in this case crucial to control operation costs and to guarantee smooth operations.

Further a major role of warehouses is the grouping of inbound and outbound flows to reduce transport costs. These roles represent the external tasks that can be assigned to a warehouse in

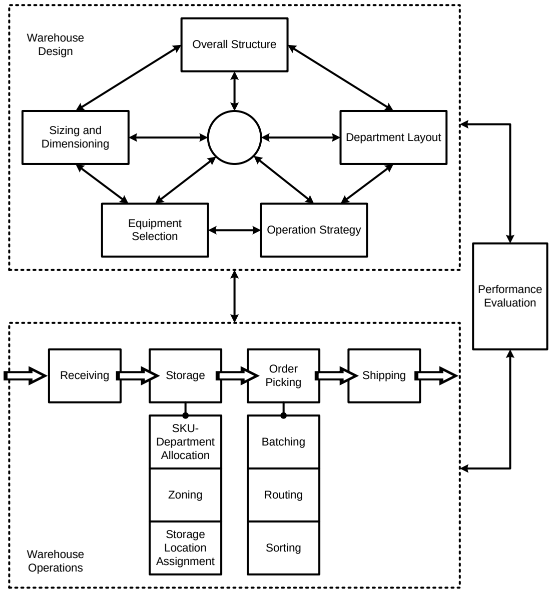


Figure 1.2: Framework for warehouse design and operation problems, Source: Gu et al. (2007)

interaction with other supply chain members. Not all tasks mentioned here are applicable to e-commerce warehouses. In this thesis, we will focus on the internal operations of warehouses, which are here included in the first bullet, namely the temporal storage of goods and the preparation of their shipment to the customer. In the following section we discuss the operations that are accompanied with this main function in more details.

1.3.2 Warehouse operations

The design of operational policies in warehouses has the overall goal to achieve cost efficient fulfillment of the assigned tasks which were introduced in the previous section. Gu et al. (2007) conduct a review on the related problems of warehouse design and operations and also illustrate their interaction. Figure 1.2 shows the authors' classification of warehouse problems.

According to Gu et al. (2007), warehouse design issues concern all related aspects that shape the warehouse in general. Those include the overall structure which includes decisions on the material flow, a separation of the warehouse into multiple departments, and their relative location in the warehouse. Also sizing questions, as the size of the overall warehouse as well as size and dimensioning of the departments, are important in warehouse design. Next, the department layout which refers to the stacking pattern, aisle orientation, length, width, and number, as well as the location of doors are problems to be decided when designing a warehouse. The authors also include the equipment selection in the design which encompasses the level of automation and the selection of storage and material handling equipment. Lastly, the operation strategy determines a storage strategy and an order picking method.

The framework of Gu et al. (2007) also illustrates the warehouse operations which control the material flow between the receiving of goods and their shipping. With respect to the incoming product flow (receiving) docks have to be assigned for arriving trucks and a corresponding dispatching strategy must be defined. The storing of goods is distinguished into the allocation of SKUs to departments, zoning, and the storage location assignment. The first determines which SKU is stored in which warehouse department, zoning describes a partitioning of departments into storage zones, and the last determines how a specific storage location for products is found and whether class-based storage is implemented. The order picking process, in turn, is distinguished into (1) batching, which groups several customer orders to one batch which is then processed by one picker, (2) routing, which defines the sequence in which the picker visits certain storage locations, and (3) sorting, at which the picked products are sorted into the individual customer orders for packaging and shipping.

In Chapter 3-5 we will consider the order batching and warehouse routing problem. Both are essential problems for the warehouse performance as they are frequently repeated operations that require a large amount of labor. According to Tompkins et al. (2010), the order picking costs represent the highest amount of the overall warehouse operation costs. Furthermore, order picking has been identified to be one of the most time consuming processes in warehouses (Dekker et al., 2004) compared to e.g., sorting and packaging. Most importantly, batching and routing allow for opportunities for a simultaneous handling of product returns and customer orders which is a major goal of this research (Chapter 3 and 4). While sorting and unpackaging of product returns usually have to be performed apart from the sorting and packaging of outgoing orders, labor costs can be reduced significantly, when product returns are integrated with orders for the batching and the picker routing. Batching problems in picker-to-parts systems, for example, are restricted by the transport capacity of the order picker. Clearly, when integrating orders with returns, this restriction can be relaxed to twice the transport capacity, if the route is designed accordingly. Besides, the two problems are interrelated. The performance of a certain batching policy is obviously dependent on the routing policy that is used to retrieve the products in the batch. In turn, the design of a method to determine short routes requires an understanding of the characteristics of the batches to be composed. We present an integrative approach in Chapter 5 to handle the combined batching and routing problem for picker-to-part systems.

1.4 Warehouses performance

Classical warehouse (design or operational) performance measures are the maximum throughput at minimum investment and operational costs, maximum storage utilization, and minimum response times (Rouwenhorst et al., 2000). With the economic changes, particularly with the movement from in-store buying to online shopping, warehouse performance becomes multifactorial (Chen et al., 2010) and encompasses also service performance components, such as product availability, on-time delivery, reliability, and correct delivery, are relevant as drivers of customer satisfaction (Xu et al., 2009; Cox and Dale, 2001). Customers have high demand with respect to service of Internet purchases. Keeney (1999) identified the fundamental objectives to be achieved in order to maximize customer satisfaction for Internet purchases. Later, many researchers adopt this framework (e.g., Kuo et al., 2009; Caruana and Ewing, 2010; Ahn et al., 2004). Keeney (1999) identifies the fundamental objectives being product quality, low cost, short time to receive a product, convenience, short time spent on purchases, privacy, shopping enjoyment, safety, and a low environmental impact. To realize these objectives Keeney (1999) classified auxiliary objectives for e-commerce companies. Some of them are system security, product availability, product information, ease of use, product variety, and reliable delivery. Inside of warehouses, operations switched to the picking of single product orders quickly, instead of processing big units for shipping to stores (Chen and Wu, 2005; Hsieh and Huang, 2011), which made service performance an essential measure for warehouses operations. One of the aims of this thesis will be to propose mathematical formulations to combine efficiency and service measurement, so that operational policies in warehouses can be judged on both. We propose examples for such measures in Chapter 3.

1.5 Planning under uncertainty

Logistics operations in general are facing a number of uncertainty sources which have an impact on policy choices and performance. One of the most intuitive examples of uncertainty sources in e-commerce is customer demand. Although for in-store buying actual customer demand is in fact also unknown, replenishment periods, inventory control, and staffing decisions are shaped by the opening hours of the store and shopping peak times during the day or for specific week days are more predictable. Online shopping has diminished this predictability. Customer orders are placed independent from opening hours, especially during weekends or evenings (Pechtl, 2003). Additionally, highly competitive environments and the uncertainty about the actions of competitors make it harder to predict customer demand. Gong and De

Koster (2011) identify uncertainty sources of warehousing systems in more detail and review models and solution approaches to deal with them. Apart from customer demand, also uncertainties of the capacity side, such as equipment failure, employee absenteeism or stock-outs, are sources of uncertainties that have to be taken into account when designing operational policies in logistics.

For the purpose of taking uncertainties into account, several research ideas are prominent. Robust optimization, for example, is an important tool to reduce the effect of specific uncertainties (Ehrgott and Ryan, 2002; Tharmmaphornphilas and Norman, 2007). Periodic review models are another example and suitable for problems in which rescheduling on short-notice is an option, so that decisions for short time periods can be made when (almost) all required information is available (Axsäter, 1993; Cachon, 2001; Tagaras and Vlachos, 2001). In Chapter 6 and 7 we discuss the literature on stochastic approaches for logistics problems in more detail. When uncertainties are high and have a high impact on performance, risk assessment and control next to the reduction of uncertainty sources can be advantageous in many situations. If, for example, demand fluctuation is very high, any optimal staff scheduling for the average demand might lead to high labor shortages on some days, to expensive over-staffing on other days. In contrast, preparing for the highest demand on all days is certainly too expensive in most situations. In such cases, an appropriate measurement of risks, which can be taken into account in an optimization and allows for efficient and risk averse planning, is useful. In the course of considering warehousing problems in e-commerce, Chapter 7 are, therefore, dedicated to the problem of risk-averse warehouse staff scheduling by means of analyzing several risk control strategies and their impact on the resulting staffing policies. We furthermore propose a tool that can help warehouse managers to take specific risks of their practical situation into account.

1.6 Special case: library warehouses

For validation purposes several warehouse cases are considered in this work of which one is a library warehouse. There are many similarities between e-commerce and library warehouses that motivate this consideration. In many libraries even the majority of material is stored in areas which are not accessible for the customers or users. Space restrictions usually prohibit that particularly older materials are stored in open shelf areas, since those would utilize more space than the tight storage of books and journals in a warehouse in which only library employees are in charge for retrieving the books in response to customer requests. In these warehouses the processing of customer requests follows the similar pattern of order placement, potentially order batching, order picking, consolidation, and eventually product return, as it is known from commercial warehouses. In this respect library warehouses share many operations regarding order picking and product return handling, with the distinctive characteristic of having a 100% return rate.

Libraries have the responsibility of collecting, organizing and preserving documents (Kohl, 2003); it is their aim to store knowledge and make it accessible over long periods of time and in the form of very different types of storage media. The amount of published material is continuously growing and even though the vast majority of new material is stored digitally, also the physically stored collection expands and libraries are usually averse to the disposal of paper copies (CHEMS Consulting, 2005). In contrast to commercial warehouses, libraries are therefore facing a continuously growing stock of several million unique items. Shared and off-site storage are currently two popular options to achieve the best utilization of library storage space (Seaman, 2004; Kohl, 2003; Chepesiuk, 1999). Next to the growing collections stored on limited space, budget limitations, warehouse versus open shelf storage, and special humidity and security requirements for some very old or valuable books lead to complex storage decisions for libraries.

Another uniqueness of library warehouses compared to commercial warehouses lies in the enormous number of unique small items that have to be managed. Dedicated storage systems or assigned policies are inevitable and, in contrast to commercial warehouses, commonly multiple copies of books are stored together rather than distributed over the warehouse. Since the storage of products at multiple locations allows for more flexible order picking, the latter is often applied in commercial settings. Moreover, library warehouses are lacking a number of characteristics that can be typically observed in commercial settings. For example, packing activities are less common since library customers usually are present to pick up their orders. Also inspection and repackaging of product returns are in library warehouses less time consuming.

However, comparing the order picking system of commercial warehouses with libraries it appears that the amount of materials to archive in libraries usually requires dense storing and thereby manual order picking as it is common in many commercial warehouses, too. In the last 30 years especially high-density storage facilities became prevalent to provide space for enormous library collections (Nitecki and Kendrick, 2001).

Even more than other warehouse situations libraries are facing service performance requirements. Mostly, no third party is involved (apart from inter library loans) for customer request processing and shipping of orders, so that the library is directly linked to its customers whose service evaluation is thereby dominantly focusing on the library operations.

Given the numerous commonalities with commercial settings study the case of a library warehouse to test the performance of our batching policy presented in Chapter 3. The results indicate that especially library warehouses can gain efficiency if they make use of the potential to integrate outgoing and incoming material.

1.7 Aims and contribution of this thesis

Given the new developments in e-commerce, the new requirements on supply chain practices and performance, and the new challenges involved, this thesis contributes to science by proposing promising solution approaches for several warehouse operational problems, namely order batching, order picker routing, and staff scheduling. Batching and order picker routing are two major operations which have a strong potential of cost savings by integrating order and return product flows. Staff scheduling in turn is a crucial issue since labor costs and planning uncertainties are typically high.

The overall goal of this thesis is to propose solution methods for warehousing problems that require a reconsideration due to two substantial challenges of e-commerce, namely product returns and demand fluctuation, which necessitate adjustments of traditional methods. To this end the thesis is aligned to the following research questions.

1. What are suitable performance measures to account for service quality next to logistics efficiency in warehouse operations?
2. How can a batching method be designed to integrate order picking with product returns processing in order to utilize existing labor capacities instead of processing returns separately?
3. If customer orders and product returns are combined in batches, what is an appropriate method to compute efficient order picking routes through the warehouse storage area?
4. Since batching and routing method performances are interrelated an integration of the two problems seems to be promising. How does an integrative approach have to be designed to solve both problems simultaneously and for which situations is this beneficial?
5. How can risk management tools be taken into account for staff scheduling problems in highly uncertain environments?

To answer these research questions we use quantitative research techniques by presenting mathematical formulations of the underlying optimization problems, propose solution approaches, and use numerical validations to evaluate the performance of our methods. Chapter 3, 4, and 5 propose solution approaches to improve and adjust the order picking to facilitate a sophisticated handling of orders and product returns. Next to a batching and a routing policy for integrated forward and return processing, we make a first step in combining interrelated warehouse operation problems into a single optimization by presenting a solution approach for integrated batching and routing. All methods are tested by numerical experiments. Our batching method, proposed in Chapter 3, is tested via simulation in comparison the practices of a library warehouse. The routing method for joint order and return processing and the integrated batching and routing solution approach are tested by means of numerical experiments. Our decision support tool for managing risks in warehouse staff planning is derived with the help of a numerical analysis and tested for a commercial warehouse case to demonstrate its applicability. Furthermore, we discuss practical implications and future research ideas to further increase order picking performance in e-commerce warehouses.

Chapter 6 and 7 of the thesis will be dedicated to an uncertainty problem that warehouses are increasingly facing in online retailing. By means of numerical experiments we analyze the applicability of various risk optimization models, their behavior in different settings, and their impact on monetary outcomes. Moreover, we develop a decision support tool that allows managers to incorporate uncertainties and risks in their staff planning procedures which we test in a case application of a Dutch commercial warehouse.

1.8 Outline of the thesis

In Chapter 2 we review the state of the art of warehouse order picking literature and thereby provide an introduction to the warehouse operation problems that are approached thereafter. Parts of this review are based on the research papers which we mention in the following. Chapter 3 relies on the article Wruck et al. (2013d) and contains two new batching methods that combine customer orders with product returns and consider the batching problem in a time-restricted context. Chapter 4 is based on Wruck et al. (2013a) in which we revisit the order picker routing problem and present a solution approach capable of finding routes to visit product pickup and delivery locations in the warehouse through building routes which fulfill the transport capacity constraint. We propose an approach to integrate the batching and routing problem in Chapter 5, which evolved in the course of the paper Wruck et al. (2013c). Chapter 6 is dedicated to the introduction of staff planning problems under uncertainty. In Chapter 7 we present a decision support tool for risk optimization in staff scheduling in warehouses. This chapter is based on Wruck et al. (2013b). Chapter 8.1 is dedicated to summarizing conclusions and implications of the research project and indicates potential future research directions for e-commerce.

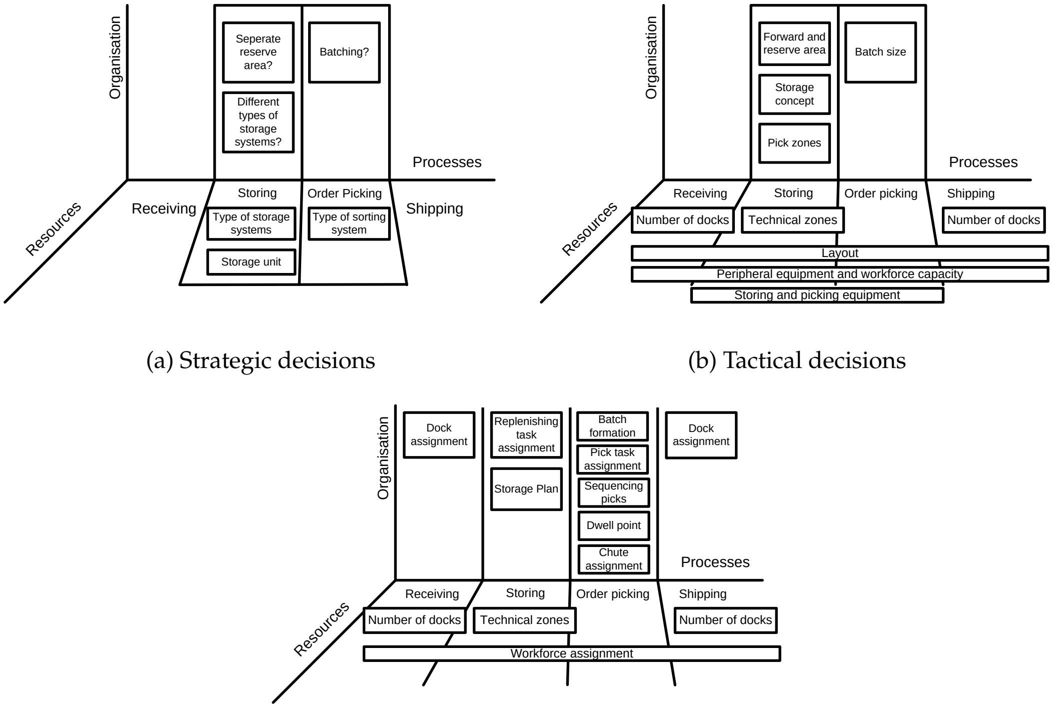
Chapter 2

Literature Review on Order Picking in Warehouses

The following three chapters of the thesis are dedicated to the redesign of warehouse processes in order to react on recent challenges that have arisen in the course of e-commerce. This chapter aims to introduce the tasks that are typically performed in warehouses to process requests of customers and we illustrate the wide range of warehouse decision problems that play an important role in practice and have, therefore, been in the focus of scientific literature.

The main functions of e-commerce warehouses are the storage of goods, the assembling of customer orders and preparation for shipping, and the inventory control (Heragu, 2008). These tasks require numerous operations within the warehouse which are typically cost intensive for the company. Efficiency, accuracy, and quickness are for that reason essential for the economic success of an online retailer. To achieve these targets many decision problems that shape a warehouse and its operations have to be considered. Sophisticated solution approaches for complex models are crucial to facilitate well-performing warehouse operations at low cost and are therefore often studied in the scientific literature. A thorough classification of warehousing problems is given by Rouwenhorst et al. (2000). The authors distinguish decision problems on three different levels, namely strategic, tactical, and operational decision level. The specific decision problems are further divided by the authors into resource, organization, or process related issues. In order to elaborate on each segment in more detail and to clarify the scope of this thesis part we first provide an overview of Rouwenhorst et al.'s classification in Figure 2.1.

Strategic level decision problems have a long-term impact and are typically accompanied with the highest costs among warehouse problems. Decisions at this level cannot be recon-



(a) Strategic decisions

(b) Tactical decisions

(c) Operational decisions

Figure 2.1: Warehouse decisions, source: Rouwenhorst et al. (2000)

sidered short- or medium-term and thus concern rather general decisions that shape the warehouse process flow design and the selection of the types of warehousing systems (Rouwenhorst et al., 2000). The process flow design determines which processes are required. The selection of warehouse system types involves, for example, the decision on an implementation of a forward/reserve storage area and whether a sorting system is required to facilitate batching.

Tactical decisions are, according to Rouwenhorst et al. (2000), for example, workforce capacity selection, but also layout decisions (Roodbergen and Vis, 2006; Hassan, 2002; Önüt et al., 2008), the storage system (Lee and Elsayed, 2005), replenishment policies (Kim et al., 2003; Zhou, 2003; Chiang and Monahan, 2005), the dimensioning of forward, reserve, and dock area, and the batch size. Also the selection of picking equipment and the level of automation (Baker and Halim, 2007; Hamberg and Verriet, 2012) are tactical decisions. The warehouse problems considered in this thesis belong to the tactical decision level as well. Decisions concerning the particular batching method that should be used and a suitable routing policy are two tactical decisions that we consider in this work. Order batching denotes the method to group a number of customer orders into sets, each of which can then be picked by one order picker. Routing in turn describes the problem of finding the shortest possible route for a given batch. Both policies have to be designed within the constraints set by the strategic level decisions. The routing policy, for example, is restricted by the storage layout and the arrangement of storages racks, but also by other tactical decisions, such as the availability and capacity of picking equipment. In general, policies selected at the tactical level can be adjusted or replaced by more suitable policies, if the outer circumstances require that. However, the tactical level encompasses medium-term decisions which, just like strategic decisions, cannot be reconsidered short-term and frequently. Related literature on order batching policies and warehouse routing methods is handled later in this chapter.

The operational level encompasses all decisions that directly affect the daily operations within the warehouse. Examples are workforce and shift assignment to employees, task assignment, allocation of incoming goods (dock assignment, storage plan, replenishment task assignment), and order fulfillment sequencing (e.g., batch formation, pick task assignment, and route determination)(Rouwenhorst et al., 2000).

Many of those decision problems are interrelated. For example, the long-term strategic decisions which specify the type of the storage, the type of sorting system (e.g., automated or manual), and the storage unit (i.e., pallet, box, or bin) have a major impact on the tactical decisions which concern, for example, equipment and workforce capacity. For example, manual order picking requires more labor, but allows to store single products very densely to achieve high space utilization. The decision to prefer one or the other is made at the strategic level. In turn, tactical decisions are influential for the decision options at the operational level, as at the latter choices are made to control the daily operations in accordance with the organizational and policy decisions made at the tactical level.

In the following literature review we focus on the warehouse problems which we approach in this thesis and refer for extensive reviews on warehousing to Rouwenhorst et al. (2000), Gu et al. (2007), De Koster et al. (2007), and Gong and De Koster (2011). A discussion of the order batching literature is provided in Section 2.1 and Section 2.2 outlines research articles dealing with order picker routing in warehouses. In Section 2.3 we discuss research that is dealing with the integrated consideration of interrelated warehousing problems and Section 2.4 concludes this chapter by summarizing the gaps in research that our review revealed.

2.1 Order batching

For the sake of clarification we begin with the definition of several terms that are often used in the batching literature. *Order batching* itself is often implemented in manual order picking systems in which order pickers are capable of transporting multiple items at the time. It describes the method to group several orders among a pool of pending orders (the *order wave*) into smaller sets which an order picker can retrieve in one single route through the warehouse. A distinction is often made between *time window batching* and *proximity batching*. The former denotes methods in which customer orders are batched based on their arrival times or due dates in order to realize short processing times and on-time delivery. The latter covers batching procedures which group items based on proximate storage locations to achieve short routes. Most literature on batching heuristics belongs to either the class of *seed-order algorithms* or *savings algorithms* (Henn et al., 2010). Seed-order heuristics construct batches step by step (e.g., Elsayed and Stern, 1983). Thereby, a (or multiple) seed order is selected to be the first order in a batch and other orders are included until a transport capacity limit is reached or until all orders are distributed among batches. Savings heuristics, in contrast, begin with a trivial batching solution (e.g., each batch consists of a single order) and batches are merged or re-sorted sequentially and according to certain rules with the objective to minimize the overall travel distance. Lastly, we can distinguish between *constructive methods* and *search-based methods*. Constructive heuristics build batching solutions sequentially (for example, seed-order methods); search-based heuristics aim to iteratively improve an initial solution by exploring solutions in a specified search region. The majority of meta-heuristic batching approaches relies on the latter.

The literature reveals that most batching policy designs are heuristic approaches, which is caused by the high combinatorial complexity of this problem. For example, the very small sized problem to batch 100 items in batches with a maximum capacity of 5 items has 79.375.495 feasible solutions. Gademann et al. (2001) prove that the decision variant of the order batching problem is NP-complete. For this reason optimal solution techniques are scarce and have not been used widely in practice, because order pools of e-commerce warehouses can be very large.

Optimal approaches for the batching problem are for that reason scarce in the literature. Armstrong et al. (1979), for example, proposed a mixed-integer model and an optimal solution approach using Benders decomposition to solve small-sized batching problems. Gademann et al. (2001) attempt to minimize the maximum throughput time for customer orders with a branch-and-bound algorithm. Gademann and Van De Velde (2005) rely on a branch-and-price algorithm and likewise solve the problem for small cases (wave size of 300 and batch size of 10 orders). Chen et al. (2005) instead use association-based clustering to maximize the similarity between customer orders within batches. A genetic algorithm, proposed in Hsu et al. (2005), might solve problems for larger sizes and provides near-optimal solutions. Henn et al. (2010) present an iterated local search approach and an ant colony optimization approach. They use S-shape and largest gap routing to evaluate the batching performance and find experimentally that significantly better solutions emerge with both proposed solution approaches rather than with constructive heuristics and basic local search methods; their iterated local search algorithm also finds solutions in shorter computation times. In line with Ho et al. (2008) and Theys et al. (2010), who combine classical warehouse-specific concepts with generally applicable meta-heuristic tools to solve the warehouse routing problem, this study supports the potential of combinations of powerful search techniques with warehouse problem characteristics. Matusiak et al. (2013) present a simulated annealing algorithm which is based on optimal precedence-constrained routing. In their experiments they obtained very small gaps to the optimal solution for batches of three customer orders and high travel distance savings for a real-life case. Finally, Gong and De Koster (2009) consider the problem of finding an optimal batch size, from a logistics and customer point of view, when the order arrival process is described by a stochastic process.

In contrast, heuristic techniques limit computation times, facilitate online implementations, and encourage practical adoptions. Beyond intuitive priority-rule procedures (e.g., first-come, first-served (FCFS)), there are more complex heuristic techniques that perform well. Ho et al. (2008) consider several constructive heuristics in their attempt to minimize the total travel distance. To test the performance of these methods the authors implement largest gap routing for a set of experiments, then combine largest gap routing with simulated annealing to determine a route for a second set of experiments. They observe that simulated annealing can help to improve routing sequences, which had been determined preliminarily with a constructive heuristic. Chen et al. (2005) design a search-based order batching policy based on association rule mining. By identifying similarities between customer orders, they aim to minimize the similarities between the resulting batches to eventually minimize labor effort. With a fixed routing method (S-shape routing) the authors find that their association rule batching method significantly outperforms the straightforward FCFS method. De Koster et al. (1999) review and test several seed-order and savings algorithms using S-shape and largest gap routing. They compare the heuristics not only with respect to the resulting travel distance and the number of batches formed but also with respect to robustness (i.e., for several warehouse layouts). They conclude that smart seed-order methods substantially outperform FCFS. Moreover, the authors propose a batching method selection tool in which they incorporate their results to identify the most suitable batching policies for a specific warehouse setting. Further, Ho and Tseng (2006) and Pan and Liu (1995) study the performance of multiple seed-order and accompanying order selection rules. Ruben and Jacobs (1999) also compare the performance of a seed-order algorithm with so-called naive heuristics that do not incorporate the storage location of items when forming batches. Other authors propose some alternatives, such as Albareda-Sambola et al.'s (2009) efficient savings algorithm that uses multiple neighborhood structures to identify better solutions. The authors test their method for several common routing methods. These neighborhood structures define neighbored solutions to be those solutions that result from, e.g., removals and reinsertions of one order into another batch to decrease the travel distance. Hwang et al. (1988) and Hwang and Kim (2005) use cluster analyses to propose other promising techniques to solve the batching problem, whereas Elsayed et al. (1993) approach the problem by determining not only the content of a batch but also the sequence of batches, in an attempt to achieve just-in-time fulfillment. Finally, the heuristic of Chen et al. (2005) aims to identify suitable batches using data mining that reveals and groups similar customer orders.

Order batching is one of the warehouse policies which allow for an integration of customer orders and product returns. Yet to maintain short response times to customer orders new models are required which account for time restrictions. By approaching the batching problem with time constraints we can facilitate returns processing together with the order picking, while performance is measured not only with efficiency indicators (as travel distance or completion time) but also with service-oriented measures such as maximum processing times. We propose suited models and solution methods for this problem in Chapter 3.

2.2 Order picker routing

Routing order pickers in warehouses aims to find a sequence of locations to be visited by the picker in order to retrieve all products in a batch. The goal is to minimize the traveled distance and the travel time, respectively. It thereby represents a special case of the Traveling Salesman Problem (TSP) with the particular characteristic of the warehouse layout, which specifies the distance metric.

Order picker routing in warehouses is well studied in research (for picking activities only). Routing problem descriptions in this context can differ in a number of warehouse characteristics such as the warehouse layout, the number and location of depots, and the restrictions on travel options through the warehouse (e.g., one-way, back and forth or effort of turns). A fundamental step toward optimal routing in warehouses has been made by Ratliff and Rosenthal (1983), who propose a polynomial time algorithm to solve the routing problem in a rectangular warehouse with one depot, multiple parallel picking aisles, and two cross aisles to optimality. Extensions of this approach are presented by De Koster and Van Der Poort (1998) for decentralized depositing and Roodbergen and De Koster (2001a) for the availability of a middle cross aisle. Next to those optimal approaches, several research studies focus on heuristic and meta-heuristic routing methods to find an appropriate trade-off between solution quality and computation time. Particularly the computational effort of large-scale problems, but also difficult warehouse layouts or the irregularity of optimal routes might impede the use of optimal routing in practice (Gu et al., 2007; Petersen and Aase, 2004). Examples of classical constructive heuristic approaches can be found in Petersen (1995), Petersen (1997), and Roodbergen and De Koster (2001b). Petersen and Aase (2004) study the effect of several classical routing methods in combination with batching and storage location assignment policies. Theys et al. (2010) consider multi-aisle warehouse layouts and evaluate the potential of more generally applicable solution techniques to the warehouse-specific routing problem. Using an adaptation of the Lin-Kernighan-Helsgaun TSP heuristic the authors demonstrate that savings up to 47% can be gained, when classical warehouse routing methods are combined with a well-designed search-based method. The authors' results and discussion also question the applicability of rigid dedicated routing methods and rather recommend a focus on solution techniques based on the specific instance rather than on the layout of the warehouse.

The warehouse routing problem with product returns, which we consider in this thesis, is closely related to a variant of the vehicle routing problem with pickup and delivery (VRPPD) in which vehicles serve customers with products that are initially located at a depot and products are picked up from customers and have to be transported to the depot. The underlying metric for VRPPDs is typically the Euclidean or Manhattan distance. The VRPPD thus describes an extension of the classical TSP, which reduces to the traditional TSP when the capacity is not restricted (e.g., Goksal et al., 2013). The objective of VRPPDs is to route multiple vehicles to minimize total travel distances. The VRPPD was first introduced by Min (1989) who proposed a mathematical problem formulation and a sequential solution procedure suitable for the distribution problem of a public library case. Mosheiov (1994) proposes two heuristics for this problem. In one approach the author determines the optimal route ignoring the capacity restriction of the vehicle first and proves that each of these routes can be made feasible by choosing another starting point of the route. The second approach is based on cheapest feasible insertion. Hernández-Pérez and Salazar-González (2004) assume that goods which are picked up at one customer can be delivered to another customer and propose a branch-and-cut algorithm to solve this problem to optimality for instances up to 75 customers. Hoff et al. (2009), Ai and Kachitvichyanukul (2009) as well as Montané and Galvão (2006) develop meta-heuristics based on tabu search and particle swarm optimization to approach the VRPPD. Bianchessi and Righini (2007) conduct a performance evaluation of several constructive, local search, and tabu search algorithms for the VRPPD and thereby focus on accuracy, computation times, and the trade-offs between those. They conclude that a local search approach with complex and variable neighborhoods performs best and is robust with respect to the diversity of instances. Nagy and Salhi (2005) present and discuss heuristic algorithms by differentiating between one and multiple depot situations. Their methods are capable of solving instances of up to 249 customers within several seconds while the results outperformed prior heuristic approaches. Li and Lim (2001) propose a tabu-embedded simulated annealing algorithm and demonstrate its applicability also for large problem instances. They define a variety of neighborhood structures and corresponding operators to identify promising neighbored solutions iteratively. We design a similar procedure to solve the integrated batching and routing problem in Chapter 5.

We contribute to this research by proposing in Chapter 4 a warehouse routing method for simultaneous order picking and return processing. In a genetic algorithm we incorporate the specific characteristics of layouts and routing options in warehouses, to facilitate a simultaneous order and return flow processing in warehouses with computation times that allow for an implementation in practice.

2.3 Integrated approaches

Many researchers recommend the integration of multiple warehouse operational problems into a single optimization problem, because of the interdependency in their impacts on order picking objectives (De Koster et al., 2007; Roodbergen and Vis, 2009; Rouwenhorst et al., 2000). For example, which options exist to build efficient routes depends on the storage layout and, in particular, on the arrangement of storage racks. Also which batching policy is possible and promising depends particularly on the order (i.e. its due date, number of items), but also on the available transport equipment and capacity of order pickers. Nevertheless, motivated by the complexity of many warehouse operational problems, most research continues to approach interrelated decision problems isolated and assumes that the corresponding other policy is fixed.

Wilson (1977) aims to find an integrative solution for the storage location and storage space assignment problem and presents a gradient search procedure to solve the resulting model. A combination of inventory management, space allocation in forward and reserve areas, and storage area layout in one single evaluation model is proposed by Malmborg (1996), who minimizes the costs associated with design and operations. Hodgson and Lowe (1982) consider lot sizing and storage space allocation at once. They aim to minimize the total material handling costs and propose a heuristic solution method for the resulting problem. Matusiak et al. (2013) propose an approach to solve the joint order batching and order picker routing problem for a general warehouse layout by using an optimal routing algorithm and a simulated annealing-based combinatorial search algorithm. Chen et al. (2010) propose an evaluative framework to identify suitable order picking policy sets to assist a warehouse manager's policy selection. Tsai et al. (2008) present an approach to integrate the batching and routing problem for a specific warehouse layout by developing a multiple genetic algorithm (GA). To find an efficient batch partitioning, they use an outer GA to identify the batch formation, and then an inner GA to evaluate the quality of these batches by optimizing the route. However, this approach is not designed to integrate product returns or situations in which multiple order pickers are available. In Chapter 5 we present an approach to integrate these issues together with the presence of customer order deadlines.

2.4 Gaps in research

With the first three research projects reported in this thesis we contribute to warehouse research in several ways:

First, with respect to batching in warehouses our review reveals a strong focus on static models, without consideration of processing time restrictions. Most studies address the batching problem in one time step, with a wave of customer orders that arrived in the past and that needs to be grouped into batches, and can be conducted in one order picking tour. The most commonly used objective is solely the minimization of travel distance. Only a few research studies incorporate deadlines. Chen et al. (2010) account for due dates in their consideration of demand characteristics when deciding on a suitable order picking policy set. Elsayed et al. (1993), for example, propose a heuristic method to solve a sequencing and batching problem for automated storage and retrieval systems to minimize earliness and tardiness of orders. Tsai et al. (2008) adopt the same objective in proposing a multiple genetic algorithm that optimizes batches and order picker routes simultaneously for manual order picking. These methods, however are not applicable to simultaneously consider order and return fulfillment, for which also time-restricted models are required in order to make use of the more flexible due date restriction of returns. Chapter 3 and Chapter 5 contain modeling approaches which take such time restrictions into account.

Second, as several researchers suggest (e.g., Ho et al., 2008; Theys et al., 2010) order picking methods should not rely solely on constructive heuristics, as there is a potential to improve performance by incorporating well-designed search methods. Also dedicated methods are usually inferior in comparison with more sophisticated methods (Theys et al., 2010). Meta-heuristic approaches represent a potential to improve order picking operations as they are capable of incorporating warehouse-specific layout characteristics in the design of search-based solution approaches (Theys et al., 2010). This can yield a strong performance, despite the complexity of the problems. Examples of search-based methods are presented in Chapter 4 and Chapter 5.

Third, most solution approaches are designed to apply for only one order picker or machine. Yet in commercial warehouses multiple order pickers are processing requests simultaneously and when on-time fulfillment is of interest, an optimization for all order pickers can be advantageous because, in this case, both batch content and sequencing is of interest for all pickers. Yet models and solution approaches for this more general case are still lacking in the literature and we present an integrative batching and routing solution approach in Chapter 5.

Fourth, with respect to product returns in warehouses in general, De Koster et al. (2002) and Stock and Mulki (2009) conducted an exploratory study among retailer warehouses to investigate how return handling is organized in practice. In both studies it is observed that returns are often still processed separately, partially due to the required special treatment (e.g., unpacking and inspection). These studies conclude that the impact of returns on warehouse operations still lacks investigation and they conclude that warehouses should be designed with respect to the amount of product returns. The main focus of the research related to warehouse operations in this thesis is dedicated to integrative approaches for forward and return product flow handling in warehouses.

Chapter 3

Batching Methods for Integrated Order and Return Processing

In e-commerce warehouses the products that customers order online are picked, sorted, consolidated, packed, and prepared for shipping. Short delivery time promises made by the company allow very limited time for these internal warehouse operations, so efficiency is crucial. Since each customer order consists of only a few single products simultaneous order picking of several customer orders in an efficient manner is desirable. Order batching is a popular tool to enhance efficiency in order picking operations. It describes the method of grouping a set of pick (and return) jobs into smaller subsets, each of which can be performed simultaneously by one order picker. Batching can help the firm organize its order picking appropriately and besides, it offers an opportunity to incorporate returns processing, which can be a strong advantage in the competitively intense e-commerce environment because it can save labor time.

In Chapter 2, Section 2.1 we discussed that, despite the promise of batching to enhance performance, the underlying optimization program is challenging to solve (Gademann and Van De Velde, 2005). This applies already to the batching problem related to order picking alone, which several research studies address (Won and Olafsson, 2012; Ho and Tseng, 2006; Ho et al., 2008). In this chapter we show that the problem also can be solved for additionally occurring product return flows, using optimization models with time constraints. Batching policies designed solely to fulfill forward product flows suffer when return rates are high, because they cannot differentiate between orders and returns and using those methods returns would be processed isolated from the order picking. Yet return processing generally enjoys more flexible time restrictions, such that the returns effectively could be included in suitable batches and be handled together with the order picking. With this approach, the picking process should not be negatively affected, but return handling still takes place.

A second focal point of this chapter is the incorporation of customer orientation into the design of efficient logistics processes. Some research recommends such multi-objective policies (e.g., De Koster et al., 2007; Heikkilä, 2002; Tang, 2010), which perform well if they achieve cost and labor efficiency, together with consumer satisfaction (Won and Olafsson, 2012). By studying the batching problem in a dynamic situation, marked by time relations between job arrivals, job fulfillment, and assigned due dates, we can use consumer-oriented objectives and a multi-objective perspective (e.g., Heikkilä, 2002; De Koster et al., 2007; Tang, 2010).

In contrast, most prior studies consider the batching problem in a static situation without processing time constraints (e.g., Albareda-Sambola et al., 2009; Ho et al., 2008; Bozer and Kile, 2008; Hsu et al., 2005; Chen et al., 2005; Gademann and Van De Velde, 2005). The few studies that incorporate time-oriented objectives include Armstrong et al. (1979), who attempt to minimize throughout times, and Elsayed et al. (1993), who assign due dates to jobs and use a penalty function for earliness and tardiness to optimize just-in-time fulfillment. Although Tsai et al. (2008) and Won and Olafsson (2012) combine logistics efficiency and customer response time, they assume that all customer orders arrived in the past and can be included in any batch. Initial steps toward a dynamic problem formulation are made by Gong and De Koster (2009) and Le-Duc and De Koster (2007) who incorporate stochastic order arrival flows. The former try to determine the optimal batch size; the latter study the impact of batching and zoning on order picking performance. In turn, our aim is to contribute to this research stream by deriving batching models that restrict the processing times allowed for jobs and thereby facilitate the efficient incorporation of product returns and a focus on on-time delivery and restricted throughput times. Those allow to find policies that match customer wishes and help firms make reliable delivery time promises. Moreover, by improving the inventory management through quick returns processing, the availability of products at any time should improve too, without overloading the warehouse. Ultimately, short, dependable delivery times are possible if the batching process is organized in accordance with expected outgoing and incoming product flows.

More precisely, we propose in this chapter two dynamic optimization models for processing forward and return flows through job batching. As the objectives of these models, we consider both logistic efficiency and consumer evaluations. Overall, we demonstrate that significant savings in time and cost can result from integrating the forward and return processes. We propose both an offline model, to evaluate solutions for a time horizon, and an online model to make batching decisions at a certain time step. In doing so, we obtain insights into how policies perform over time and how to use information about fluctuations of the job arrival process. The models apply to any warehouse setting that receives a significant amount of product returns and uses manual order picking, such as e-commerce retailers, catalog companies and library warehouses. For our numerical experiments, we use the case of a library warehouse. Similar to traditional retailers, libraries are confronted with return flows that are in this special case as large as their order flows and therefore provide a well-suited application example.

The remainder of this chapter is structured as follows: We develop our two dynamic batching models in Section 3.1 and outline advantageous performance measures. In Section 3.2 we present a solution approach for each model. Section 3.3 is dedicated to numerical examples, in which we show the effect of simultaneous order and return job processing as well as the behavior of the batching process using different measures. Our conclusions are in Section 3.4.

3.1 Modeling approach

We propose two dynamic optimization models for integrated forward and return flow handling that can account for customer response times and on-time delivery by employing the different processing time restrictions for order and return jobs and can achieve short travel distances. One dynamic model pertains to an offline context and describes an entire time horizon, such that orders and returns arrive over time, must be fulfilled over time, and have corresponding due dates within a time horizon. Available solution techniques, such as savings, neighborhood search, or genetic algorithms, cannot be applied easily in this case, because they require feasible solutions to begin, and for models with time constraints, such solutions are not trivial to find. As Gong and De Koster (2009) did, the offline model can be used to identify general rules that shape the batching procedure by identifying optimal batch sizes, tour start intervals, and conditions for including product returns.

The second dynamic model is designed for online application and makes batching decisions at a certain time step in periodic review; its description is similar to traditional descriptions (wave picking) of a large set of customer orders that must be grouped into batches. In contrast with most classical problem formulations, our online model starts with a decision about whether a batch should be formed at all, considering all currently pending jobs and their instantaneous throughput times. If a batch is formed, not all pending jobs must be included, but they might be postponed to the next review period. For these optimization models, we formulate suitable performance measures that should be maximized.

3.1.1 Offline model

As we noted, many studies have formulated optimization models for the batching problem for a wave picking situation, which is common in warehouses with many customer orders. In an offline setting, several waves arrive at the warehouse at certain time steps within the time horizon, so the solution of an offline optimization model can be useful to identify rules at the tactical level that match the regular job flow. For example, the size of the waves, the jobs they contain in terms of arrival times and types (pick or return jobs), and the time steps at which the waves arrive at the warehouse are tactical decisions that can be made with the help of an offline model. Taking the throughput times of all jobs into account, it might also be wise to detect and handle peaks in the job arrival process and bottlenecks in the fulfillment process. Moreover, for tactical decisions, such as the number of order pickers or the purchase of transport equipment, an offline solution might reveal the effects of changes on the utilization of capacities and service quality. However, our offline model does not include waves of jobs; rather, we consider the job arrival process itself, across the entire time horizon, and aim to compose batches for single order pickers. The model can be applied for one or more order pickers. It is restricted only by the due dates of jobs and the durations of the tours. That is, each job must be fulfilled before its due date, and a new tour of an order picker cannot start before his or her previous tour has ended.

To develop the model, let $\mathcal{T} = \{0, \ldots, T\}$ be a discrete time horizon. Jobs can arrive at any time step in $\mathcal{T}$, and an order picker's tour can start at the specific time steps $\{t_1^*, \ldots, t_I^*\} = \mathcal{T}^* \subset \mathcal{T}$. We assume that jobs are unit-sized and that each job consists of a single product to be picked or returned. Customer requests containing multiple products are treated as separated jobs. We define the set of all products in the warehouse by $\mathcal{N} = \{1, \ldots, N\}$, and $\mathcal{M} = \{1, \ldots, M\}$ is the set of order pickers. Products can be ordered or returned by customers. All products to be picked must be delivered to an i/o point in the warehouse (the depot) by the order picker; all returned products are initially located at the depot and must be transported to their storage location in the warehouse. Thus, an order picker's route always starts and ends at the depot. We define the set $\mathcal{S} = \{0, 1\}$, where 0 denotes that a product is ordered and 1 denotes that a product is returned. With the help of these notations, we interpret each job j as an element of the set

$$\mathcal{J} \subset \mathcal{N} \times \mathcal{T} \times \mathcal{S}.$$

Thus, a job $j \in \mathcal{J}$ can be represented as the 3-tuple $j = (n_j, t_j^{arr}, s_j)$, where $n_j \in \mathbb{N}$ specifies the product to be picked or returned and its location; $t_j^{arr} \in T$ is the arrival time of the job; and $s_j \in S$ the type of the job.

Two types of variables describe the tour composition. First, for each time step in which a tour can start $t^*_i \in \mathcal{T}^*$ and the order picker $m \in \mathcal{M}$, we define the binary variable $r^{t^*_i,m}$ as follows:

$$r^{t^*_i,m} = \begin{cases} 0, & m \text{ does not start a tour at } t^*_i \\ 1, & m \text{ starts a tour at } t^*_i \end{cases} .$$

Second, to describe the inclusion of a job $j \in \mathcal{J}$ in a tour of order picker m starting at time step t^*_i, we use a binary variable $inc_j^{t^*_i,m}$. Each job has a certain due date t_j^{due} , depending on its arrival time and its type. For each job j with $t_j^{arr} \leq t^*_i < t_j^{due}$ we thus can define

$$inc_j^{t^*_i,m} = \begin{cases} 0, & j \text{ is not included in the tour of } m \text{ starting at } t^*_i \\ 1, & j \text{ is included in the tour of } m \text{ starting at } t^*_i \end{cases} .$$

Because logistics efficiency and customer service simultaneously influence warehouse process optimization, it is difficult to define warehouse performance; highly efficient policies might not imply good service performance and vice versa. We therefore develop several appropriate objectives and determine how to combine these targets.

Minimizing travel distances is a popular efficiency-oriented objective. The travel distance to fulfill one batch depends on the jobs it contains and the storage location of each product. An optimization model to minimize the total travel distance of all formed batches can be stated as follows

$$\min \sum_{i=1}^{I} \sum_{m=1}^{M} L_{J^{t^*_i,m}} \tag{3.1}$$

subject to:

$$\sum_{m=1}^{M} \sum_{i=1}^{I} inc_j^{t^*_i,m} \;=\; 1 \quad \forall j \in \mathcal{J}, \tag{3.2}$$

$$inc_j^{t^*_i,m} - r^{t^*_i,m} \;\leq\; 0 \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M},\ \forall t^*_i \in \mathcal{T}^*, \tag{3.3}$$

$$r^{t^*_{i+k},m} - (1 - r^{t^*_i,m}) \;\leq\; 0 \quad \forall k:\ t^*_i < t^*_{i+k} \leq t^*_i + D_{J^{t^*_i,m}},\ \forall m \in \mathcal{M}, \tag{3.4}$$

$$\text{feas}_{J^{t^*_i,m}} \;=\; 1 \quad \forall m \in \mathcal{M},\ t^*_i \in \mathcal{T}^*, \tag{3.5}$$

$$\text{lead}_j + t_j^{arr} - t_j^{due} \;\leq\; 0 \quad \forall j \in \mathcal{J}, \tag{3.6}$$

where $J^{t^*_i,m}$ is the batch formed at time step t^*_i and fulfilled by order picker m. Furthermore, $L_{J^{t^*_i,m}}$ denotes the corresponding tour length to fulfill batch $J^{t^*_i,m}$, and $D_{J^{t^*_i,m}}$ is its duration. We define $L_{J^{t^*_i,m}} = D_{J^{t^*_i,m}} = 0$, if there is no tour starting at t^*_i which is conducted by order picker m (i.e., $r^{t^*_i,m} = 0$). $\text{feas}_{J^{t^*_i,m}}$ is a function to evaluate the feasibility of a batch for transport capacity, assuming the implemented routing method is fixed. It is defined as follows:

$$\text{feas}_{J^{t^*_i,m}} = \begin{cases} 1, & r^{t^*_i,m} = 0 \\ 1, & r^{t^*_i,m} = 1 \text{ and } J^{t^*_i,m} \text{ is feasible} \\ 0, & r^{t^*_i,m} = 1 \text{ and } J^{t^*_i,m} \text{ is infeasible} \end{cases} .$$

The objective function (3.1) expresses the sum of all tour lengths of batches which are built at the time steps $\{t_1^*, \ldots, t_I^*\} = \mathcal{T}^*$ and fulfilled by order pickers $\{1, \ldots \mathcal{M}\}$. Constraint (3.2) requires each job to be fulfilled exactly once. With Constraint (3.3), we require that a job can be included only in time steps in which batches are formed. Constraint (3.4) prohibits a new tour start for an order picker if his or her previous tour has not ended. Regarding the number of jobs in a batch, more flexibility is possible with incorporated returns, so the number of jobs is not necessarily limited to the maximum transport capacity. However, transport capacity cannot ever be exceeded during a tour. Constraint (3.5) expresses that all formed batches are feasible with regard to transport capacity within the tour. The throughput time lead_j (lead time) of a job j is the time-lag between the arrival of an order at the warehouse and its delivery to the depot. If j is contained in the batch $J^{t_i^*,m}$, its lead time can be derived by

$$\text{lead}_j = t_i^* - t_j^{arr} + D_{J^{t_i^*,m}} \quad \text{for } j \in \mathcal{J} : inc_j^{t_i^*,m} = 1.$$

Thus, Constraint (3.6) requires the fulfillment of each job before its due date.

To maintain a certain service level and similar to (3.1), we can develop objectives that are more preferable from the customers' perspective. The maximum throughput time for order jobs, for example, should be as short as possible within warehouse capacity constraints. Let $\mathcal{J}_{\text{ord}}$ be the set of order jobs. As an alternative to (3.1), the objective of minimizing the maximum lead time can be stated as

$$\min \max_{j \in \mathcal{J}_{\text{ord}}} \text{lead}_j. \tag{3.7}$$

For the consumer, the throughput time of ordered products is of greater interest, and this objective would reduce the number of returns in a batch. We then need an appropriate constraint to find solutions that incorporate the fulfillment of return jobs. Instead of Constraint (3.6), which limits the allowed lead time for jobs in general, we add a constraint for objective (3.7) that restricts the lead time specifically for return jobs, such as

$$\text{lead}_j \leq t_j^{\text{due}} \quad \text{for } j \in \mathcal{J} : s_j = 1. \tag{3.8}$$

Another important driver of customer satisfaction is on-time product delivery. When the seller mentions an exact delivery time, the customer expects the product to be available at exactly that time. Tardiness thus can significantly affect customers' evaluations of service quality. This also holds for settings in which third party suppliers are involved, whose scheduling is highly depended on on-time fulfillment within the warehouse. Furthermore, in very busy settings, Constraints (3.6) and (3.8) might cause infeasible optimization programs. Thus, we provide another service-oriented objective, namely, minimizing tardiness, and introduce another binary variable that describes the lateness of a job $j \in \mathcal{J}$. For a job j that is included in a batch

$J^{t_i^*,m}$ we define

$$\text{late}_j = \begin{cases} 0, & t_i^* + D_{J^{t_i^*,m}} \leq t_j^{\text{due}},\ j \in \mathcal{J} \\ 1, & t_i^* + D_{J^{t_i^*,m}} > t_j^{\text{due}},\ j \in \mathcal{J}. \end{cases}$$

With the help of this variable, we can express the objective of minimizing the number of late deliveries as

$$\min \sum_{j \in \mathcal{J}} \text{late}_j. \tag{3.9}$$

For this objective, the lead time restrictions in (3.6) and (3.8) are not needed. However, for both objectives (3.1) and (3.7), another adequate constraint appears to be necessary to control for exceeded due dates. Instead of the constraints (3.6) or (3.8), we might tolerate late deliveries, but we target a solution with a certain maximum number of delays. Let $\text{late}_{\text{max}} \in N_+$ be the maximum number of late deliveries accepted. They can be limited by requiring

$$\sum_{j \in \mathcal{J}} \text{late}_j \leq \text{late}_{\text{max}}. \tag{3.10}$$

Because efficiency and service orientation usually play a simultaneous role, even as they might come in conflict, a beneficial trade-off is required for practice. To achieve and evaluate such a compromise, we propose an objective that combines the two preceding objectives, namely, travel distance and maximum throughput time.

$$\min \left(\sum_{h=1}^{H} \sum_{m=1}^{M} L_{J^{t_i^*,m}} + \lambda \cdot \max_{j \in \mathcal{J}_{\text{ord}}} \text{lead}_j \right). \tag{3.11}$$

With this objective, the overall travel distance and maximum lead time can be minimized simultaneously. The parameter λ is used as scaling factor to account for differences in units, as well as to weight the desired relevance of the different objective variables against each other if they are in conflict.

With the help of this offline model, we can identify rules that shape the batching policy, adjusted to the regular job flow, such as average batch sizes and tour start intervals. It can also support tactical decisions about required labor and equipment. Moreover, we can analyze the effects of different optimization objectives on the batch formation. To refine this batching procedure for daily fluctuations, we next propose an online optimization model, in which we define multiple test time steps in the time horizon. It is possible to test online whether a batch should be formed and, if so, which jobs it should contain, given information about currently pending jobs. Insights on the periods between test time steps can be obtained by analyzing solutions of the offline model.

3.1.2 Online model

In deterministic online models, information about incoming job flows is not known in advance. Decisions are made only on the basis of information about the past. In contrast with other studies though, we do not formulate the online model for a wave picking situation. We consider a model in which all jobs that arrived before a certain test time step and remain unfulfilled get taken into account. A batch does not have to be formed at each test time step. Rather, a prior decision reveals whether an order picker tour is required, according to the number of currently pending jobs and their instantaneous lead times. This model and the solution approach we present subsequently are more suitable in smaller settings (with fewer jobs) and for a highly varying job arrival process. If the average batch size and average tour start intervals are known, adjustments for exceptionally busy days might be useful.

The objectives of an online approach cannot be the optimization of characteristics that evolve over time, as in (3.1). Rather, the decisions aim to maximize performance of a suitable measure at one time step. To develop the model, we first let $t_i^* \in \mathcal{T}$ be a fixed point in time. At t_i^* there is a set of pending order and return jobs $\mathcal{J}^{t_i^*} = \{j_1, \ldots, j_K\} \subset \mathcal{J}$ that arrived in the past and have not been fulfilled in any previous tour. If order picker m is available at time step t_i^*, the first decision is whether a batch should be formed. In this case, we define clear conditions for the value of the variable $r^{t_i^*,m}$ (see the offline model formulation on page 35). We propose a decision based on the time gap between t_i^* and the arrival times $t_{j_k}^{arr}$, $k = 1, \ldots, K$ of the pending jobs and the number of pending jobs K, respectively:

$$r^{t_i^*,m} = \begin{cases} 1, & \text{there is a job } j_k \in \mathcal{J}^{t_i^*} : \; t_i^* - t_{j_k}^{arr} \geq \tau_{\max} \\ 1, & K \geq J_{\max} \\ 0, & \text{otherwise.} \end{cases},$$

where $\tau_{\max} \in R_+$ is the maximal tolerated time gap between the arrival time of a job $t_{j_k}^{arr}$ and the current time step t_i^*, and $J_{\max} \in N_+$ is the maximum number of pending jobs allowed. In case $r^{t_i^*,m} = 1$, a batch is formed that contains some currently pending jobs.

Next, to evaluate the quality of a potential batch, we need a suitable performance measure. First, we provide a measure that includes both tour length and the number of jobs contained in the batch (batch size). For any potential batch $J^{t_i^*,m} \subset \mathcal{J}^{t_i^*}$ we define the length-based performance of the batch $\mathcal{P}^{\text{length}}_{J^{t_i^*,m}}$ by

$$\mathcal{P}^{\text{length}}_{J^{t_i^*,m}} = \alpha \cdot |J^{t_i^*,m}| - L_{J^{t_i^*,m}},$$

where $\alpha \in R_+$ is a control parameter, similar to the scaling factor λ in (3.11). That is, it weights the importance of the batch size and the tour length, respectively. With the help of this length-based measure, we propose the following optimization model to be solved at time step t_i^*:

$$\max \mathcal{P}^{\text{length}}_{J^{t_i^*,m}} \tag{3.12}$$

subject to:

$$\text{feas}_{J^{t_i^*,m}} = 1 \quad \forall m \in \mathcal{M},\ t_i^* \in \mathcal{T}^*, \tag{3.13}$$

where Constraint (3.13) assures the feasibility of the resulting tour, as introduced in the offline model.

Another performance measure, beyond tour length, also accounts for the lead times of order jobs. A method that evaluates a batch on the basis of throughput times alone would simply form batches with the longest pending jobs (first-come, first-served); instead, any incentive to account for the efficiency of the resulting tour should be combined with service-based measures. Here, we must recognize that the lead times of jobs not contained in the current batch cannot be determined exactly. However, these jobs must have a negative impact on service-based performance measures, so we introduce a penalty value $pen \in R_+$ that is added to the instantaneous lead time of jobs that are not included and that redefines the (partially estimated) lead time of pending jobs in t_i^* by

$$\text{lead}_j = \begin{cases} t_i^* - t_j^{arr} + D_{J^{t_i^*,m}}, & j \in J^{t_i^*,m} \\ t_i^* - t_j^{arr} + D_{J^{t_i^*,m}} + \text{pen}, & j \in \mathcal{J}^{t_i^*} \backslash J^{t_i^*,m} \end{cases}.$$

Using this equation, we can define a service-based measure to evaluate the throughput time of all pending jobs, according to the maximum lead time of included orders and the sum of the estimated lead time of excluded jobs. That is,

$$tpt_{J^{t_i^*,m}} = \max_{j \in J^{t_i^*,m}_{\text{ord}}} \text{lead}_j + \sum_{j \in \mathcal{J}^{t_i^*} \backslash J^{t_i^*,m}} \text{lead}_j,$$

where $J^{t_i^*,m}_{\text{ord}}$ denotes the set of included order jobs in batch $J^{t_i^*,m}$. A performance measure that combines time and length indicators can thus be defined for any batch $J^{t_i^*,m} \subset \mathcal{J}^{t_i^*}$ by

$$\mathcal{P}^{\text{comb}}_{J^{t_i^*,m}} = |J^{t_i^*,m}| + \beta \cdot |J^{t_i^*,m}_{\text{ord}}| - \gamma \cdot tpt_{J^{t_i^*,m}} - \delta \cdot L_{J^{t_i^*,m}},$$

where β, γ, and δ again are scaling factors. The resulting optimization model to be solved at t_i^* we obtain

$$\max \mathcal{P}^{\text{comb}}_{J^{t_i^*,m}} \tag{3.14}$$

subject to constraint (3.13).

3.2 Solution approaches

In this section, we propose solution approaches for the dynamic batching models from Section 3.1, which are applicable to each objective that we proposed. Depending on the underlying objective, we define a performance measure to evaluate the quality of a single batch. When solving the offline model, this measure is used to construct batches iteratively; it instead is used to identify optimal batches at one time step when the online model is to be solved. For the description of the solution approaches we assume that the travel speed of the order pickers is constant and we do not include storage ans retrieval times. Extensions of the solution methods to account for both, varying travel speed and the incorporation of storage and retrieval times, can easily be made.

3.2.1 Seed-order algorithm for offline application

With the help of the previously defined offline model, we propose a new seed-order algorithm (see Section 2, page 24). The structure of this heuristic is based on the sequential selection of seed orders. The accompanying order rule is controlled by the performance measure, according to the underlying objective and the corresponding constraints (Section 3.1.1), so the performance measure must be selected in advance. The selection of accompanying orders is limited to the set of jobs for which the arrival time is close to the arrival times of jobs already included in the batch. Iteratively, those jobs join the batch which optimize the performance measure with respect to the corresponding tour start, tour end, and due dates. Thus we ensure the fulfillment of Constraints (3.2) - (3.6) as defined on page 35. If a batch is complete or no more jobs can be included, a new seed order is selected, defined as the first arrived and unfulfilled job. If this seed order cannot be fulfilled on time because of a late tour end for the previous batch, a swapping method incorporates the job in the previous batch. If swapping alone cannot facilitate an on time fulfillment of the earliest arrived, not included job (i.e., the new seed order) jobs with later due dates have to removed from the batch to realize an earlier tour end. In detail, the algorithm is as follows:

Seed-order algorithm for dynamic order and return job batching

1. *Initialization:* Initialize the set of jobs, their due dates and the performance measure:
   Let $\{j_1, \ldots, j_N\} = \mathcal{J}$ be a set of all jobs. We assume that the jobs are sorted according to their arrival time, i.e.,

$$t_{j_1}^{arr} \leq t_{j_2}^{arr} \leq \cdots \leq t_{j_N}^{arr}.$$

Let $tp_{\text{ord}}^{\max}$ and $tp_{\text{ret}}^{\max}$ be the maximum allowed throughput times of order jobs and return jobs, respectively. Calculate the due dates for job j_k by

$$t_{j_k}^{due} = \begin{matrix} t_{j_k}^{arr} + tp_{\text{ord}}^{\max}, & \text{for } s_{j_k} = 0 \\ t_{j_k}^{arr} + tp_{\text{ret}}^{\max}, & \text{for } s_{j_k} = 1 \end{matrix} .$$

Select a performance measure to evaluate the quality of a batch $B_{i,m}$, which matches the given objective of the model (Section 3.1.1). For example, for the distance-based objective (3.1), we would use the measure

$$\mathcal{P}(B_{i,m}) = \alpha \cdot |B_{i,m}| - L_{B_{i,m}}$$

to evaluate the quality of a batch $B_{i,m}$.

Denote length and duration of a batch $B_{i,m}$ by $L_{B_{i,m}}$ and $D_{B_{i,m}}$, respectively.

Select a tolerance parameter t_{tol} which defines the search interval for jobs to be included.

2. *First seed order selection:* Select the first seed order and define the search interval:

   Select the first arrived job j_1 to be the first seed-order, assign it to an order picker m, and denote the first batch $B_{1,m)} = \{j_1\}$. Identify all jobs with arrival times in the time interval

$$SI_{B_{1,m}} = \left[\min_{j_k \in B_{1,m}} t_{j_k}^{arr}, \min_{j_k \in B_{1,m}} t_{j_k}^{due} - t_{\text{tol}} \right] .$$

   The parameter t_{tol} is required, because the tour cannot start before the arrival of the last included job and must be fulfilled before the due date of the first arrived job. Next, calculate the instantaneous performance of the batch $\mathcal{P}(B_{1,m})$.

3. *Accompanying orders:* Select accompanying orders iteratively:

   Identify a best-fitting job for the batch by solving the optimization program

$$\max_{j_k : t_{j_k}^{arr} \in SI_{B_{1,m}}} \mathcal{P}(B_{1,m} \cup \{j_k\}) - \mathcal{P}(B_{1,m}) \tag{3.15}$$

   subject to:

$$\begin{aligned} \max_{j_l : t_{j_l}^{arr} \in SI_{B_{1,m}} \cup \{j_k\}} t_{j_l}^{arr} + D_{B_{1,m} \cup \{j_k\}} &\leq \min_{j_l : t_{j_l}^{arr} \in SI_{B_{1,m}} \cup \{j_k\}} t_{j_l}^{due}, & (3.16) \\ \text{feas}_{B_{1,m}} &= 1. & (3.17) \end{aligned}$$

   In this model, the performance difference between the original batch and the potential new batch is maximized. The first constraint ensures that the tour is completed before the maximum due date of each included job, to fulfill Constraint (3.6). The second constraint requires the feasibility of each batch in terms of transport capacity (3.5).

(a) If a solution has been found, include the corresponding job in the batch and repeat step 3.

(b) If no solution can be found, proceed.

4. *Batch start times and tour ends:* Calculate tour start and end of a batch:
The tour start of a batch $brt_{B_{i,m}}$ is the arrival time of the latest arrived job included in the batch. The corresponding tour end $te_{B_{i,m}}$ is the sum of start time and tour duration.

5. *New seed order:* Select a new seed order:
Let $B_{i,m}$ be the last batch built in previous iterations. Among all remaining jobs select the job with the earliest arrival time to be the next seed order.

   (a) If there is an unfulfilled job j_k, test order picker availability.

      i. If an order picker m is available to fulfill job j_k or

$$\max\left(t^{arr}_{j_k}, \max_i te_{B_{i,m}}\right) + tol \leq t^{due}_{j_k} \tag{3.18}$$

      denote the new batch with $B_{i+1,m}$ and proceed to step 2.

      ii. If no order picker is available to fulfill job j_k on time, proceed with step 6.

   (b) If all jobs are included in batches, stop.

6. *Swapping method:* A swapping method between included jobs and the leftover job gets applied until the earliest arrived, un-included job fulfills condition (3.18).

The optimization model in step 3 allows for a decrease in batch performance, which is necessary because performance might decrease by including jobs into a batch with, for example, travel distance as a performance indicator. The test in step (3a) thus evaluates the performance decrease and mandates that if the decrease exceeds a certain tolerance, the corresponding job is not included in the batch.

3.2.2 Online model solution approach

In an offline view, online solution approaches always lead to a heuristic solution, because even optimal decisions in each review period cannot necessarily produce an optimal solution for the entire time horizon. The complexity of the batching problem in general, and considering that in practice online solutions are required quickly, leads us to propose an approach with several relaxed restrictions by the test time step selection (e.g., order picker availability), such that an optimal (maximum) value of the performance measure $\mathcal{P}$ can be found at the current time step. We defined several suitable performance measures for the online situation in Section 3.1.2. The online solution approach for a time step t^*_i can be described as follows:

Online solution algorithm for integrated forward and return flow batching

1. *Order picker availability:* Check the availability of an order picker.
   (a) If an order picker is available, proceed.
   (b) If no order picker is available, proceed with step 6.

2. *Initialization:* Identify the currently pending jobs $j_k \in \mathcal{J}^{t_i^*}$ and their due dates $t_{j_k}^{due}$, $k = 1, \dots, K$.

3. *Batch formation test:* Decide whether a batch should be formed by considering the number of pending jobs and their instantaneous lead times.
   (a) Form a batch if at least one job is urgent (i.e., there is a $j_k \in \mathcal{J}^{t_i^*} \; : \; t_i^* - t_{j_k}^{arr} \geq \tau_{max}$) or there are more than J_{max} jobs pending. Denote the urgent jobs by $j_{k_i}^{urg}$, $i = 1, \dots, I$ and proceed with step 4.
   (b) If there are neither urgent nor too many jobs pending, proceed with step 6.

4. *Formation of a central batch:* Include as many as possible urgent jobs $j_{k_i}^{urg}$,$i = 1, \dots, I$ in the batch. If no urgent jobs are pending, the central batch is empty. Otherwise,
   (a) If all urgent jobs form a feasible batch (in terms of transport capacity), proceed with step 5.
   (b) If not, determine the position within the tour in which the capacity problem occurs and remove the corresponding job from the central batch $(I = I - 1)$. Repeat step 4 for the reduced batch.

5. *Optimal filling up:* Include pending jobs in the central batch that maximize the performance measure. Solve the optimization program

$$\max_{J^{t_i^*,m} \subset \mathcal{J}^{t_i^*}} \mathcal{P}(J^{t_i^*,m})$$

   subject to:

$$j_{k_i}^{urg} \in J^{t_i^*,m}, \quad \forall i = 1, \dots, I.$$

   where $\mathcal{P}$ denotes one of the performance measures defined in Section 3.1.2.

6. *Termination:* Remove the batch from the set of pending jobs and determine the next test time step.
   (a) If no order picker is available, define the next test time step as $t_{i+1}^* = t_i^* + \text{post}_{op}$.

(b) If the central batch has to be reduced, define the next test time step as $t^*_{i+1} = t^*_i + \text{post}_{cb}$.

(c) Otherwise, use the regular interval length between test time steps: $t^*_{i+1} = t^*_i + il$.

To control for tardiness, the constants post_{op} and post_{cb} are incorporated; they reduce the time until the subsequent test time step if not all urgent jobs could be fulfilled or if no order picker was available. However, the values for τ_{max} and J_{max} must be defined in advance; we cannot always prevent late deliveries. We show with examples in Section 3.3 that with a well-considered parameter choice, the number of late deliveries is small. Furthermore, we incorporated a function to accelerate optimization in step 5 in situations in which no urgent jobs but more than $2J_{max}$ jobs are pending. In this case, the constructed central batch contains a small number of jobs, located close to each other in the warehouse, order to reduce the combinatorial complexity of the optimization in step 5.

3.3 Numerical and practical validation

We study the effects of our solution approaches for the two dynamic models using numerical examples derived from a real-life case. We simulate the online solution approach over a time horizon of one day. With these examples, we can demonstrate the importance of efficient return flow processing, judged according to logistic efficiency and customer response time. We perform our experiments using transaction data for a library warehouse and compare the batching solutions of the dynamic methods with the order picking practices of this library. Library warehouses are a good option for our experiments, because they feature 100% return rates. For ease of explanation, we assume only one order picker is available for each experiment. As we explained previously, both models are suited for cases with more than one order picker though. Moreover, the solution approaches can be used for any kind of storage location assignment policy implemented in the warehouse. The library warehouse consists of three floors, each with 24 parallel aisles. A parallel aisle can be switched only at a cross aisle in the middle of the floor. Stairs to switch between floors are located next to the fourth and the twentieth parallel aisle. The entrance, which is considered the depot, is located at the fourth parallel aisle of the ground floor. The storage location assignment in this library warehouse reflects clustering in topics; location assignment is not based on customer order frequency.

Because performance of the order batching problem and the routing problem are highly interrelated, we used a fixed routing policy for order picking (Ratliff and Rosenthal, 1983) in the dynamic models. This routing method is optimal for one-type situations (only picking or only returning activities). Thus, when considering two types of jobs simultaneously, the route might be infeasible in terms of transport capacity, resulting in an infeasible batch. In our dynamic batching models, another batch would be selected.

Gademann et al. (2001) highlight that computation efforts depends not only on the number of jobs to fulfill in total but also on the batch size, or the number of jobs contained in one batch. Including picking and return jobs simultaneously in this problem doubles the potential batch size. We consider on average 100 jobs (picks and returns) per day, and a customer with multiple jobs can be split up over more than one batch. The transport capacity is limited to five items in the dynamic models, which results in a maximum batch size of ten jobs. We derived 14 test instances related to transaction data of 14 days varying patterns in the number of jobs, from 50 to 144. The percentage of return jobs in these examples varied from 38% to 62%.

We implemented the solution approaches in C++; the experiments were conducted on an Intel Core 1.33 GHz processor. All results of the offline seed-order algorithm were calculated within a few seconds. The calculation of the online model simulation was performed in less than five minutes for each sample day.

In Table 3.1, we provide the results for the offline and online solution method, with two different objectives: total travel distance or the combination of travel distance and the maximum throughput time of order jobs (objectives (3.1) and (3.11), Section 3.1.1). We compare the results based on the indicators total travel distance and maximum throughput time of order jobs. The results of the offline solution (Panel a) strongly confirm that the maximum throughput time of order jobs can be reduced significantly (46% on average), by incorporating it as an objective. However, this decrease is obviously accompanied by longer tours (14% on average). The results from the online solution in Panel b show that in most instances, the maximum throughput time again can be decreased by incorporating it into the objective. However, in contrast with the offline solution, short throughput times cannot be assured in online settings, as some results show. Very efficient batches in the beginning of the time horizon might lead to postponements, especially of return jobs, when using the combined performance measure. Bottlenecks at later periods can thus cause high throughput times for later orders.

With Table 3.2 we indicate the importance of a differentiation between orders and product returns. For the distance-based objective without any differentiation, the possible batch size is limited to the transport capacity of the order picker, so the result indicates how existing methods, designed for one-type job batching, perform when they must address not only order and but also return jobs. Our experiments in the real-life case thus show the savings possible through combining the order and return job processing. There is an essential improvement opportunity available in a distinction of order and return jobs. First, a differentiation of the

Sample day	#jobs	td* objective		Combined objective	
		td* (m)	Max tpt* (min)	td* (m)	Max tpt* (min)
a. Offline solution approach					
1	50	634	77	650	8
2	62	803	78	974	54
3	72	773	82	853	35
4	84	936	85	1094	45
5	87	1061	79	1309	59
6	93	1035	79	1177	27
7	100	1118	80	1415	37
8	110	1307	88	1528	60
9	114	1401	86	1509	67
10	116	1407	84	1503	35
11	123	1427	83	1756	43
12	131	1380	86	1621	68
13	137	1396	87	1720	62
14	144	1532	85	1794	34
b. Online solution approach					
1	50	554	78	737	51
2	62	908	74	869	73
3	72	752	89	852	80
4	84	843	74	945	78
5	87	1046	69	1240	71
6	93	944	84	1072	103
7	100	1050	90	1206	73
8	110	1279	80	1382	95
9	114	1363	85	1572	75
10	116	1220	96	1641	89
11	123	1538	84	1592	84
12	131	1549	87	1663	89
13	137	1626	98	1586	79
14	144	1623	97	1732	89

* td - travel distance, tpt - throughput time.

Table 3.1: Total travel distance and maximum throughput time for orders

Sample day	#jobs	Offline solution		Online solution	
		td* (m)	Max tpt* (min)	td* (m)	Max tpt* (min)
1	50	963	71	720	110
2	62	1310	81	1058	128
3	72	1313	77	963	140
4	84	1332	77	1308	147
5	87	1621	84	1324	138
6	93	1686	77	1297	169
7	100	1906	83	1265	163
8	110	2239	80	1761	130
9	114	2193	81	1647	149
10	116	2430	81	1700	132
11	123	2454	79	2024	159
12	131	2471	83	2020	148
13	137	2461	81	1988	134
14	144	2700	81	2138	151

* td - travel distance, tpt - throughput time.

Table 3.2: Undifferentiated job types

type leads to significantly shorter total travel distances (44%). Second, regarding the maximum throughput time of order jobs using the offline solution algorithm (restricted to 90 minutes), all the solutions produced enormously increased travel distances. From the simulation of the online solution approach for the entire day, we can determine that many small batches lead to not only longer travel distances but also substantially larger throughput times. Using the type differentiation, we can reduce the number of small batches that might contain only urgent returns and delay order job processing.

Finally, we show in Table 3.3 the performance that currently practiced job fulfillment attains in the library warehouse. In practice, the order picking process is separated to three picking and three return rounds per day. All order jobs that arrive before a pick round are included in the next pick batch, as are all returns included in the return batch. A cart transports the items, and the size of each batch can be greater than 10. No optimal routing is used; instead, the order picker fulfills the jobs in a batch separately for each floor, using an elevator (next to front stairs) to transport the cart. Within one floor, the order picker likely follows an optimal route, though to confirm that our results did not depend on different routing methods, we also conducted experiments with an optimal routing policy. Differences between the actually used (non-optimal) routing method and optimal routing were small and had only minimal negative effects on the total travel distance, likely because in practice the batches can be very large. Longer travel distances instead result from separated fulfillment of pick and return batches.

Sample day	#jobs	td* (m)	Max tpt* (min)
1	50	1373	167
2	62	1507	140
3	72	990	145
4	84	1313	151
5	87	1593	150
6	93	1438	124
7	100	1695	200
8	110	1878	233
9	114	1593	241
10	116	1534	180
11	123	2402	247
12	131	1986	236
13	137	2075	239
14	144	2218	221

* td - travel distance, tpt - throughput time.

Table 3.3: Job fulfillment in practice

In the large order pick round, the order picker travels through most of the locations of the warehouse; conducting returns in a second round leads to an unnecessary revisiting of many locations. The results, compared with the dynamic models, demonstrate that on average it is possible to obtain 31% savings in travel distance by integrating the pick and return rounds. The maximum throughput time of orders is obviously high for the processing in practice, because batches are formed statically at fixed time steps, independent of the arrival process.

3.4 Concluding remarks

This chapter deals with batching policies that integrate order and return flow processing which can lead to significant improvements in warehouses operational performance. Two dynamic models, in offline and online contexts, support the design of a more efficient job batching process through a problem formulation with processing time restrictions. We developed two suitable solution approaches and validated them with the help of numerical examples and in comparison with a real-life case. The seed-order algorithm proposed for the offline model, provides good heuristic solutions in a very short computation time and thus could be used in large warehouse settings to support tactical decisions that reflect the given job flows. It also can help design an online approach. Future research is required to analyze to what extent the method performs well to provide good start solutions for metaheuristic search approaches. In contrast, the online optimal batches must be found quickly, so the amount of jobs determines the applicability of this approach. Computation times increase with the batch size, which can be twice as high as the transport capacity if we integrate order and return job handling. For the warehouse case we study, we find that the proposed online solution approach delivers very good results in acceptable computation times.

In e-commerce warehouse cases it should be remarked that additional sorting and consolidation effort is required which might be higher, if customer orders are conducted in multiple batches. Also the administrative tasks to complete returns processing might be necessary. The trade-off between the time savings achieved by integrated batching and the additional time effort required for sorting process and the like has to be balanced for a specific case. Nevertheless, time and cost savings can be possible also in commercial warehouse by integrating forward and return flows. Future research to study the precise case of commercial warehouses might be required to analyze to which extent the methods are transferable.

Furthermore, we have provided insights into how consumer-oriented performance measures can be incorporated into the design of batching policies and the affect they have on total travel distances. Multi-objective performance measures help significantly shorten response times and also lead to the formation of efficient batches. With our dynamic seed-order algorithm, savings algorithms and other powerful meta-heuristics become possible tools for solving the batching problem with time constraints.

This research also shows the strong interdependency of batching and routing policies in warehouse order picking and we demonstrated the potential benefits of integrating forward and return flows for operational policies. In the following chapter we therefore consider the corresponding routing problem which integrates order and return job processing to make the best use of the transport capacity of the order picker and picking device. Chapter 5 will then focus on an integrative approach.

Chapter 4

Order Picker Routing with Product Returns

The most labor intensive operation in warehouses is the order picking process. The picking is for many product categories still processed manually, since high storage space utilization mostly prohibits automation. Especially when batching is implemented and many locations have to be visited in the warehouse to pick the orders of multiple customers, routing policies that lead to efficient picking routes are highly important, as labor time is expensive. When in addition to customer orders also product returns are contained in the batch, new warehouse routing policies are required. Our aim in this chapter is to design a method capable of incorporating product returns in the picking routes in order to make best use of the available capacities instead of processing product returns separately.

The warehouse routing problem under study is the following: Order pickers travel through the warehouse to visit a number of locations at which they pick or return products. Each route starts and ends at a central depot at which all products to be returned are initially located, and at which all picked products have to be deposited. As the transport capacity of the picking device is limited and routes start with a certain load of returns, this implies that the order picker cannot visit locations to pick products at some points in the route. Our objective is to design a method to define routes in that each pickup and return location is visited, that never exceed the transport capacity, and that minimize the traveled distance.

Also already indicated in Chapter 2 routing problems in warehouses form a special case of the Traveling Salesman Problem (TSP). The classical TSP (i.e., for Euclidean norms) is known to be NP-complete (Papadimitriou, 1977). For some specific warehouse layouts, such as single block and two block warehouses, the corresponding routing problem can be solved in polynomial time (e.g., Ratliff and Rosenthal, 1983; Roodbergen and De Koster, 2001a). With an increasing number of cross aisles the routing problem approaches a Manhattan TSP, which has been shown to be NP-complete as well (Papadimitriou, 1977). With the integration of product returns in the picking route, a new challenge is added to the problem. Then, this case is related to the vehicle routing problem with pickup and delivery (VRPPD). The transport capacity of the picking device has to be respected at every point of the route. In this case, the warehouse routing problem appears to be highly complex already for warehouses with fewer cross aisles. Besides, in practice, routes often have to be determined online and very short computation times are thus desirable. As these routes have to be determined frequently and periodically in commercial settings, short order picking routes can realize major cost savings, and sophisticated solution methods are of high practical relevance.

In this chapter we propose a genetic algorithm (GA) to solve the warehouse routing problem with pickups and returns. Thereby we make use of the potential of well-designed search techniques and combine it with the specific characteristics of the warehouse routing problem. The GA is suited to identify short routes in computation times which are acceptable for online applications. Moreover, it is independent from the specific warehouse layout and can therefore be widely applied for various layouts. We demonstrate that using the GA near-optimal - and often even optimal - solutions can be determined and we perform an analysis to identify the best mix of returns and picking requests in order picker routes.

The structure of this chapter is as follows. In Section 4.1 we give a detailed problem description and introduce an ILP formulation of the problem under study. Section 4.2 is dedicated to the illustration of the GA, and we summarize and discuss the results of our numerical experiments in Section 4.3. We make concluding remarks on this chapter in Section 4.4.

4.1 Problem description

To model the warehouse routing problem with pickup and return we use the following notation:

$\mathcal{P}$	Set of pickup locations
$\mathcal{D}$	Set of return locations
n	Number of locations to be visited
$\mathcal{N}$	Set of locations to be visited including the depot; ($\lvert\mathcal{N}\rvert = n+1$)
c_{ij}	Travel distance between the locations $i, j \in \mathcal{N}$
$p_i \geq 0$	Volume to be picked at location $i \in \mathcal{P}$
$d_i \geq 0$	Volume to be returned at location $i \in \mathcal{D}$
$q > 0$	Transport capacity

We denote the set of all locations to be visited by $\mathcal{N} = \{0, \ldots, n\}$, where 0 corresponds to the location of the depot. The sets $\mathcal{P} \subset \mathcal{N}$ and $\mathcal{D} \subset \mathcal{N}$ denote the subsets of locations with pickup and return requests, respectively. Potential locations with both pickup and return requests are treated as two separated locations with a distance equal to zero. Each route of an order picker starts and ends at the depot and the storage location and volume of each product to be picked or returned at this location is known beforehand. We assume that each set of products, which in total does not exceed the capacity, can be transported at once, i.e. products are assumed to have uniform sizes. Each location with pickup or return requests has to be visited and multiple visits of the depot, which could elude the capacity constraint, are not permitted. Our decision variables are

$x_{ij} \in \{0,1\}$	Is 1, if the order picker travels along arc (i, j), 0 otherwise
$y_{ij} \geq 0$	Load already picked and transported along arc (i, j)
$z_{ij} \geq 0$	Remaining load to be returned and transported along arc (i, j)

The binary variable x_{ij}, for $i, j = 0, \ldots, n$, corresponds to the decision whether location j is visited directly after location i was visited, or not. By means of the non-negative variables y_{ij} and z_{ij} we control the transported load along each arc (i, j) with $i, j \in \mathcal{N}$ in the network. Obviously, if the locations i and j are visited subsequently it has to be assured that the sum of the load picked so far (y_{ij}) and the load which still has to be returned (z_{ij}) does not exceed the transport capacity.

The order picker routing problem with pickup and return in warehouses can be formulated as

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} \tag{4.1}$$

subject to

$$
\begin{aligned}
\sum_{i=0}^{n} x_{ij} &= 1 && \forall j \in \{0,\dots,n\} && (4.2)\\
\sum_{j=0}^{n} x_{ij} &= 1 && \forall i \in \{0,\dots,n\} && (4.3)\\
\sum_{j=0}^{n} y_{ij} - \sum_{k=0}^{n} y_{ki} &= \begin{cases} p_i & i \in \mathcal{P} \\ -\sum_{i=1}^{N} p_i & i = 0 \\ 0 & i \in \mathcal{D} \end{cases} && && (4.4)\\
\sum_{j=0}^{n} z_{ij} - \sum_{k=0}^{n} z_{ki} &= \begin{cases} -d_i & i \in \mathcal{D} \\ \sum_{i=1}^{N} d_i & i = 0 \\ 0 & i \in \mathcal{P} \end{cases} && && (4.5)\\
y_{ij} + z_{ij} &\leq qx_{ij} && \forall i,j \in \{0,\dots,n\} && (4.6)\\
y_{ij},\ z_{ij} &\geq 0 && \forall i,j \in \{0,\dots,n\}\\
x_{ij} &\in \{0,1\} && \forall i,j \in \{0,\dots,n\}
\end{aligned}
$$

The objective function (4.1) represents the total travel costs to be minimized when traveling a complete route. Constraints (4.2) and (4.3) assure that each location is visited exactly once. With Constraints (4.4) and (4.5), we control for the currently transported volume between any pair of locations i and j. Constraint (4.4) requires that the volume of any location i is picked, if $i \in \mathcal{P}$, and that all products were picked, when the order picker returns to the depot (i.e., $i = 0$). Constraint (4.5) states the condition that the entire volume to be returned at location i is returned , if $i \in \mathcal{D}$, and that all items were returned at the end of the route. Constraint (4.6) limits the transported load between each pair of locations i and j to the transport capacity of the order picker.

4.2 Genetic algorithm

Our literature review revealed a clear suggestion to make use of more generally applicable meta-heuristic approaches in combination with warehouse-specific constructions (Theys et al., 2010; Ai and Kachitvichyanukul, 2009; Bianchessi and Righini, 2007). Genetic algorithms are a class of meta-heuristics that are often applied for complex and large-scale optimization problems. Inspired by evolutionary theory, they rely on the generation of a set of solutions, an initial population, which is iteratively altered by crossover and mutation functions. At each step the quality of a solution is evaluated and influences the probability of its attributes to survive in the next generation. For the warehouse routing problem at hand genetic algorithms appear to be a well-suited tool for a number of reasons. First, an initial population (i.e., a set of sequences to visit all locations) is very easy to generate. Second, with a sophisticated crossover definition we can facilitate the inheritance of attributes of well-performing solutions. These attributes correspond to partial sequences of locations in our case. Third, the mutation of individuals (i.e., single solutions in a population) is usually implemented to avoid that the GA stops at local minima. Here, mutations also allow us to incorporate warehouse-specific characteristics by developing mutation operators which make use of the warehouse layout information.

The general procedure of a GA can be described as follows. After creating the initial population, the objective value of each individual solution within the population is calculated. In the mutation phase some randomly selected individuals are mutated by means of one or multiple mutation operators. Then, in the crossover phase a number of solution pairs are selected based on probabilities that are defined with respect to their quality. Better performing individuals are more likely to be selected for crossover. Each selected solution pair is recombined in a certain manner to create a new solution. In the last phase, the selection phase, the solutions which build the following generation, i.e. the new population, are determined. Thereby, a certain percentage of the by crossovers created solutions and the mutated individuals replace a number of solutions in the current population. Doubling of individuals in the same generation should be avoided in order to maintain diversity in the population. The algorithm stops, when a fixed number of iterations has been performed. In the following we describe each step of our proposed GA for the warehouse routing problem with pickup and return in detail.

4.2.1 Initial population

Diversity of solutions in the populations is essential in genetic algorithms in order to allow for a broad search perspective. In preliminary experiments on which we elaborate later in the chapter, we observed that especially for large instances a good initial population is advantageous for high quality solutions. We, therefore, do not construct the initial population completely random, but employ a variant of the nearest neighbor heuristic by incorporating parameterized regret based random sampling, as for example used in Fang and Wang (2012). The classical nearest neighbor heuristic is a deterministic constructive procedure in which at each step, beginning at the depot, the route is extended by including the closest unvisited location (Dorigo and Gambardella, 1997). Obviously, this procedure would always lead to the same routing sequence. Hence, to obtain diversity in the population, we incorporate a degree of randomness in the sense that subsequent locations are selected randomly, but based on probabilities according to their distance from the current location. Let i denote the current location. The next visited location $j \in \mathcal{N}$ is selected either out of all remaining return locations, if the currently transported load up to location i equals the transport capacity, or out of all remaining unvisited locations otherwise. For those locations $j \in \mathcal{N}$ that do not violate the capacity constraint we define the priority v_j to visit the location subsequently by

$$v_j = \max_{k \in \mathcal{N}_\mathrm{u}} c_{ik} - c_{ij}, \tag{4.7}$$

where $\mathcal{N}_\mathrm{u} \subset \mathcal{N}$ denotes the set of unvisited (feasible) locations and c_{ij} denotes the distance between two locations i and j. We use the priority of a location to define a regret value r_j for not selecting a location by

$$r_j = v_j - \min_{k \in \mathcal{N}_\mathrm{u}} v_k. \tag{4.8}$$

Last, we define the probability p_j for selecting the location j by

$$p_j = \frac{(r_j + 1)^\alpha}{\sum_{k \in \mathcal{N}_\mathrm{u}} (r_k + 1)^\alpha}, \tag{4.9}$$

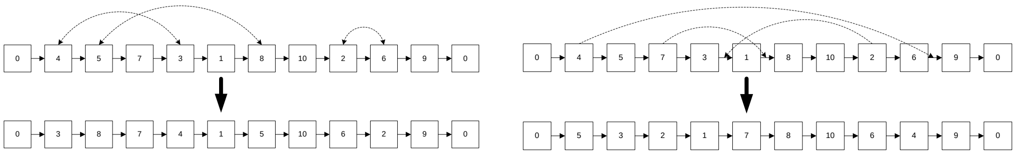
where $\alpha \in [0, \infty)$ determines the degree of randomness incorporated. Thereby, $\alpha = 0$ provokes a completely random selection of locations. With an increasing value of α the route construction becomes more deterministic, while prioritizing closer locations over distant ones. In our numerical experiments we discuss the impact of various selections of the control parameter α on the solution quality.

4.2.2 Mutation

As noted earlier the role of mutation operators for our solution approach is twofold. First, mutations are essential in genetic algorithms to prevent that the procedure stops at local minima (Hong et al., 2000). With a crossover function alone it might be harder to find good solutions. Second, in the mutation phase we also incorporate the warehouse-specific characteristics when defining mutation operators, such as sorting partial sequences in an intuitive order with respect to the warehouse layout. Taking these specifications into account facilitates that the search procedure of the genetic algorithm can become more efficient and focused (Theys et al., 2010).

A mutation rate $p_m \in (0, 1)$ determines the percentage of the population size to be altered by mutation operators in each iteration. A mutated individual thereby replaces its original in every case, even when the resulting performance is worse than the original individual's performance. A prioritization of individuals to be selected for mutation is not made; each individual is equally likely to be selected. For the selected individuals one out of four different mutation operators is randomly selected. Each of those mutation operators is now described.

For the first two mutation operators, illustrated in Figure 4.1 we define an integer-valued control parameter $S > 0$ which determines the intensity of change between the original individual and the mutation. The first mutation operator creates a mutated individual by swapping



(a) Operator 1: exchange

(b) Operator 2: removal and reinsertion

Figure 4.1: Mutation operator 1 and 2 for $S = 3$

location pairs within the sequence. The number of swaps made is given by S. While small values of S result in mutations which are still very similar to the original individual, larger values of S will produce more alteration of the original. The second mutation operator is designed in a similar manner. It forms a new individual by removing a randomly selected location from the sequence and reinserting it at another again randomly selected position in the sequence. The number of removals and reinsertions made by this mutation operator is again controlled by the parameter S. This design suggests a careful definition of the value of S. Clearly, a low value for S could cause a decrease of diversity in the population, which in turn might lead to local optima. In contrast, any too large value for S might impede that mutations survive in the population over a number of iterations, as in the selection process better performing solutions must be prioritized to a certain extent. We elaborate on well-suited selections of the control parameter S in our numerical experiments.

The mutation operators 3 and 4 are illustrated in Figure 4.2. By means of these mutation operators we make use of the specific warehouse characteristics and provoke that all locations to be visited within an aisle are sorted according to an intuitive routing sequence. With the third mutation operator any aisle which contains locations to be visited is randomly selected. All locations in this aisle are sorted in such a way that results in traversing the aisle from one aisle end to the other, i.e. locations to be visited in this aisle are sorted by their location in the aisle. The direction of sequencing is randomly selected. The resulting partial sequence containing all locations of one aisle is inserted in the complete routing sequence at any point of the original individual at which the corresponding aisle was visited before. By implication, all locations within this aisle are removed from the sequence at all other positions. The fourth mutation operator is designed in a similar manner. Here, again one aisle which contains locations to be visited is randomly determined. In contrast to the third mutation operator, all locations in this aisle are sorted in such a way as the aisle is entered from one side by the picker, all returns are conducted on the picker's way in the aisle, the picker turns at the farthest pickup or return location, and all pickup requests are fulfilled on the picker's way to leave the aisle. Again,

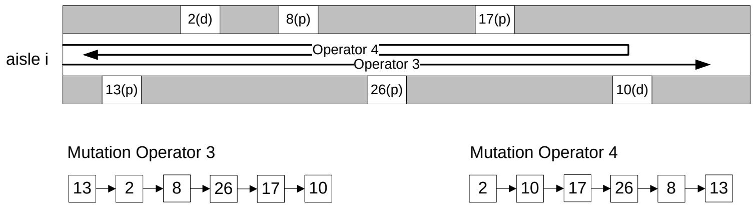


Figure 4.2: Mutation operator 3 and 4 ((p)=pick, (d)=deliver)

the side on which the picker enters and leaves the aisle is randomly selected according to a uniform distribution.

Mutations are used to facilitate the creation of attributes which might not be contained in the individuals of the population yet. The four operators designed here therefore facilitate that the GA can escape from local minima, which can here be interpreted with solutions that result from the best combinations of attributes which are currently existent in the population. Next to that, we apply with the help of operator 3 and 4 a kind of local focus within single aisles to create potentially promising attributes.

4.2.3 Crossover

The crossover phase aims to inherit well-performing attributes to the next generation. To do so, the algorithm determines the objective value of each individual solution in the current population, including mutated individuals. Its objective value determines an individual's probability to be selected as parent for a crossover. Subsequently, based on those probabilities two parents are chosen. The creation of a new individual (child) from two parents is similar to the method of Kazarlis et al. (1996) and can be described as follows. Any location to be visited is randomly selected. Up to this location all locations are inserted in the new individual in the same sequence as in the first parent. All remaining locations are added to the new individual in the sequence in which they appear in the second parent. The crossover procedure is illustrated in Figure 4.3. Doing so, two potentially well-performing individuals are re-combined in a way that maintains advantageous partial sequences. Obviously, the crossover as explained so far would allow only little variation in the beginning of sequences, as the first part (of random length) of an individual would always be adopted by the children from the first parent. To prevent this effect we apply the crossover procedure either by starting at the beginning or at the end of the routing sequence. The selection of the direction is determined randomly beforehand for each newly created child.

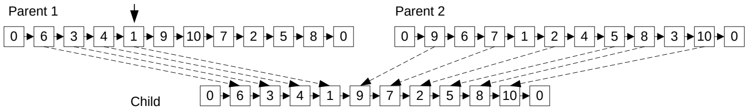


Figure 4.3: Crossover

4.2.4 Selection

In the selection process we determine the new population, consisting of individuals of the previous population, some mutations, and some of the new individuals created by crossovers. The degree of change from one population to another needs to be considered carefully. While too few new children entering the population could decelerate the algorithm and require more iterations, too many entering new solutions might harm the diversity of the population and lead to local minima. The number of new individuals, which enter the population is controlled by a parameter $s \in (0, 1)$, which represents a percentage of the population size. Along the entire procedure we maintain a constant population size, so that only the best newly created individuals enter the population by replacing the worst solutions of the current population. We refer to the following section, in which we discuss the impact of this survival rate s on solution quality. Furthermore, we allow for a limited number of infeasible solutions in each population. The mutation as well as the crossover procedure can lead to the creation of infeasible solutions. However, those solutions might still have valuable attributes (in terms of well-performing partial sequences) and should not generally be excluded from the population. In the discussion of our numerical experiments we elaborate on suitable values to control the number of tolerated infeasible solutions.

4.3 Numerical experiments

We describe our experimental setting in Section 4.3.1. To calibrate the control parameters of the GA, such as suitable population size, mutation, and survival rate, we conducted experimental tests and report on the corresponding results in Section 4.3.2. Thereafter we provide in Section 4.3.3 insights in the performance and applicability of the GA by comparing its computational results with the optimal solutions and two constructive heuristics for larger instances. Lastly, in Section 4.3.4 we aim to identify the best policy to integrate product returns with customer orders by determining the best possible division of pickups and returns for order picker routes in practice.

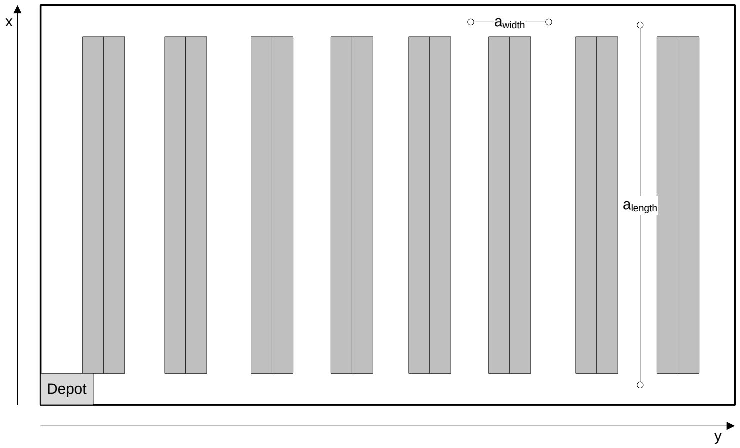


Figure 4.4: Rectangular warehouse

4.3.1 Experimental setting

As noted above, a strength of our solution approach is the applicability for a variety of warehouse layouts. Particularly, the GA is independent of the length, alignment, and number of storage aisles. Our experiments we therefore conducted for a commonly used rectangular warehouse layout with two cross aisles and the front and back side of several parallel aisles of equal length (De Koster and Van Der Poort, 1998). This layout is depicted in Figure 4.4.

The configuration of our experimental design is similar to previous literature on warehouse routing (e.g., De Koster and Van Der Poort, 1998; Roodbergen and De Koster, 2001b). The length of the parallel aisles varies for different experiments, the aisle width is set to 2.5 meters. Pickup and return locations are assumed to be located solely in the parallel aisles and not in the cross aisles. Further, as we consider relatively narrow aisles, we assume that order pickers can reach the left and right hand side of an aisle without traveling additional distances. Cross aisles can be used to switch between aisles. Moreover, each aisle can be traversed in both directions. Locations in the aisles are assigned accurate to 0.1 meters. Capacity and transported load are measured for unit sized products. For each location there is one unit of products to be picked or returned. Adjustments for more detailed or varying weights can easily be made.

4.3.2 Control parameters

In preliminary experiments we restricted ourselves to the standard form of the problem, which describes the situation in which both pickup and return load coincide with the transport capacity (Mosheiov, 1994). Each pick list consists of 50% pickups and 50% returns, while the trans-

Route constructions for initial population	2000
Degree of randomness α	20
GA iterations	500
Population size	600
Number of crossovers	600
Survival rate s	0.33
Mutation rate p_m	0.05
Mutation intensity S	1
Allowed infeasible solutions	0.05

Table 4.1: Control parameters

port capacity equals the number of pickups and returns, respectively. Obviously, these are the most difficult problems and we thus optimize our control parameters for such instances. For the same reason, the preliminary experiments are conducted for larger batches with 60 and 100 locations to be visited in total.

For the creation of an initial population, we determined the best results with constructing 2000 routes. α was thereby set to 20 and the best of those constructed routes formed the initial population. The high value of α leads to the creation of an initial population of high quality, which turned out to be superior to an initial population of lower quality and higher diversity. Regarding the number of GA iterations, we found in all tests that the solutions did not improve significantly after 500 iterations, so that in the remaining experiments the number of iterations is fixed to 500. Moreover, we observed that the GA performs best when a large population size is maintained across the entire procedure. This effect might be explained by the size of instances and thereby with the number of feasible solutions, which requires a large population. Nevertheless, given the tight restrictions on computation times in practical applications we used a population size of 600 individuals, while the population size remains constant across the GA iterations.

Given this population size, we found that the generation of 600 children is appropriate. The best performing survival rate s, as defined in Section 4.2.4 on page 59 has been found to be 33%. Selecting more children fails to focus on promising areas of the search region, whereas selecting fewer children apparently leads to too fast convergence and therewith potentially to adherence in local solutions. With respect to the mutation control parameters we noticed that a mutation rate of $p_m = 5\%$ provides the best results. Higher mutation rates harm the search process, since mutated individuals always enter the population. In contrast, lower mutation rates limit the search region, so that solely the inheritance of attributes within the initial population can improve the solutions.

The mutation intensity S is best set at 1 location swap, larger mutation intensities yield solutions too far away from the good individuals and thereby cannot survive in the population. For the selection phase we observed slight improvements when allowing for 5% infeasible solutions in each generation compared to the exclusion of infeasible solutions. Higher limits for the amount of infeasible solutions in the population however could not improve the solutions and too high values obviously even harmed the quality of the results. An overview of the best performing selection for control parameters is provided in Table 4.1.

4.3.3 Results

The sample data to study the performance of our algorithm is illustrated in Table 4.2 which provides an overview of the warehouse size (i.e., number and length of aisles) and the instance size (i.e., transport capacity of the order picker and batch size, which describes the number of locations to be visited in one route). Clearly, there is a difference in the problem difficulty between hard instances in which the transport capacity is fully utilized (i.e., batch size = $2\cdot$ capacity) and other instances with more flexibility. It consists of 18 scenario sets each of which consisting of 100 instances. These instances consist of an equal number of return and pickup locations to be visited. The transport capacity and the warehouse size vary.

We provide a comparison with optimal solutions, obtained with CPLEX, for the scenario sets $1-6$. However, optimal solution procedures substantially suffer from increasing instance sizes. While the computation times for the scenario sets 1 and 2 were a few seconds only, for scenario set 5 results were computed in approximately 30 minutes. For the larger instances of the sets $7-18$ no optimal solutions could be obtained in realistic computation times. For those scenarios the constructive cheapest feasible insertion heuristic and an adapted variant of S-shape routing serve as upper bounds. Cheapest feasible insertion (CFI) describes the method which initially creates a routing sequence consisting of the depot and two random locations only. All remaining locations to be visited are included in this routing sequence one after another at the position in the routing sequence at which they cause the least additional travel cost and do not violate the capacity constraint. As a second heuristic approach we adjust traditional S-shape routing to make it applicable for pickup and return requests. When using S-shape routing the picker entirely traverses each aisle containing locations to be visited. Thereby each second aisle is traversed in reverse direction. Adjusted to the case of pickup and return requests, pickup locations at which the capacity would be exceeded are skipped and the order picker returns to the aisles with these locations on his or her way back to the depot. Also in this second round aisles are traversed entirely. The results of our comparison are summarized

Scenario Set	Aisle length	# Aisles	Capacity	Batch Size
1	12	7	10	20
2	12	7	15	20
3	12	7	15	30
4	12	7	20	30
5	12	7	20	40
6	12	7	30	40
7	12	7	25	50
8	12	7	35	50
9	32	15	30	60
10	32	15	40	60
11	32	15	35	70
12	32	15	45	70
13	32	15	40	80
14	32	15	50	80
15	32	15	45	90
16	32	15	55	90
17	32	15	50	100
18	32	15	60	100

Table 4.2: Sample datasets

in Table 4.3 for the scenario sets $1 - 6$, and in Table 4.4 for the scenario sets $7 - 18$.

All methods were programmed in C++ and experiments were performed on an Intel Core 2, 2.99 GHz processor. Solutions of the heuristics S-shape and CFI routing were obtained within 1 second. The GA delivered results within 10 seconds (scenario sets 1 and 2) and 2 minutes on (scenario set 17). The optimal solutions obtained with CPLEX took between a few seconds for the scenario sets 1 and 2 and 30 minutes on average for scenario set 5.

The results provide a number of insights on the performance of our algorithm. First, Table 4.3 shows that we obtained very small gaps to the optimal solutions, especially for scenario

Set	Optimal	GA	GA to optimal	CFI	CFI to GA	S-Shape	S-shape to GA
1	103.05	103.57	0.50%	133.89	20.76%	136.27	23.53%
2	101.51	101.72	0.20%	109.34	6.77 %	124.75	18.09%
3	111.06	112.56	1.36%	156.64	25.89%	142.47	20.37%
4	109.32	109.39	0.06%	122.19	10.06%	129.75	15.61%
5	109.02	114.72	5.30%	170.93	30.98%	145.16	20.39%
6	109.06	112.22	2.98%	129.66	13.14%	130.04	13.64%

Table 4.3: Results of scenario sets 1 - 6 (distances in meter)

sets in non-standard form. When the transport capacity is not completely utilized by pickup or return requests, more flexible routing is possible, which facilitates the GA to maintain a diverse population of feasible solutions and good quality.

Also the results of larger instances in comparison with CFI and S-shape routing show strong improvements. Compared to the CFI method improvements of 20.14% were obtained on average over all scenario sets. Further, the GA led to 15.20% shorter routes than S-shape routing on average. We also observe that the gap between CFI routing and the GA solutions increases with the number of locations if we consider and non-standard form instances as well as the small and the large warehouse layout separately. The reason for this is the rigid nature of the cheapest feasible insertion heuristic. The special metrics in warehouses might require short detours when building order picking routes in order to avoid that the picker has to return to distant aisles at the end of a route. CFI does not account for this by building routes in a constructive way. In contrast, the GA is capable of keeping a variety of partial sequences of locations in the population and explores various combinations of those, which eventually leads to better results. With respect to the comparison of our approach with S-shape routing we observe the opposite effect of the gap. Again, separated into standard and non-standard form scenarios as well as the small and the large warehouse layout, we find that the gap between the GA solutions and S-shape routing decreases with an increasing instance size. This effect can be explained partially by the decrease of performance of the GA with is identifiable in comparison with the optimal routes. However, if the number of locations to be visited increases relative to the overall number of locations in the warehouse, certainly passing nearly all locations in the warehouse might become inevitable to build a route that visits all required locations. S-shape routing might thus be a good method to conduct very large batches in general. If additionally the capacity is not completely utilized, detours might not be necessary, which explains the relatively small gaps between GA and S-shape solutions for the scenario sets 14, 16, 18. Nevertheless, overall we could demonstrate that the GA leads to significantly better solutions for all scenarios tested here.

4.3.4 Batch composition experiments

To give insights in the practical application of our solution approach we conducted a second set of experiments to analyze the most suitable composition of pickup and return requests in the batches. Clearly, the total number of return requests will typically be lower in e-commerce settings than the number of picking request. Warehouses in online retailing confront return rates of 18 to 74% (Mostard et al., 2005), depending on the product category and return oppor-

Scenario Set	GA	CFI	Gap CFI to GA	S-Shape	Gap S-shape to GA
7	118.04	176.28	30.97%	145.64	18.27%
8	114.88	134.37	14.15%	131.00	12.31%
9	511.38	688.75	25.06%	642.63	20.14%
10	503.08	573.16	12.08%	570.95	11.75%
11	525.08	703.44	24.42%	667.60	18.50%
12	516.14	598.99	13.63%	578.48	10.70%
13	540.65	756.93	27.53%	652.23	16.67%
14	531.86	625.88	14.85%	582.47	8.65 %
15	552.28	783.39	28.73%	659.91	15.84%
16	541.77	649.64	16.41%	587.59	7.73 %
17	557.49	794.24	28.66%	658.68	15.05%
18	548.54	674.51	18.40%	585.72	6.34%

Table 4.4: Results of scenario sets 7 - 18 (distances in meter)

Case	# Batches	# pick batches	# return batches	# mixed batches	Picks / returns per mixed batch
1	160	80	0	80	30 / 30
2	160	64	0	96	30 / 25
3	160	40	0	120	30 / 20
4	160	0	0	160	30 / 15
5	176	0	16	160	30 / 12
6	208	0	48	160	30 / 6
7	240	160	80	0	-

Table 4.5: Batch composition cases

tunities. The dataset that was used for the following experiments consists of 7200 locations to be visited in total, of which one third (i.e., 2400) are return requests. The goal of these experiments is to determine the best combination of pickup and return requests in single batches. We aim to find out whether the returns should be distributed only over a few batches, or whether an even distribution of returns over all picking batches is more advantageous. For these experiments the larger warehouse layout with 15 aisles and an aisle length of 32 meters was used. The locations of the 7200 pickup and return requests where assigned randomly with a uniform distribution. Here, the capacity of the picking device is set to 30. Seven cases of batch compositions were computed, which are characterized in Table 4.5. The cases vary between a full integration of pickup and return requests in one half of all batches, while the second half of batches contains pickup requests only (case 1) and a completely separated processing of pickup and return requests in 240 batches, each of which containing 30 requests (case 7).

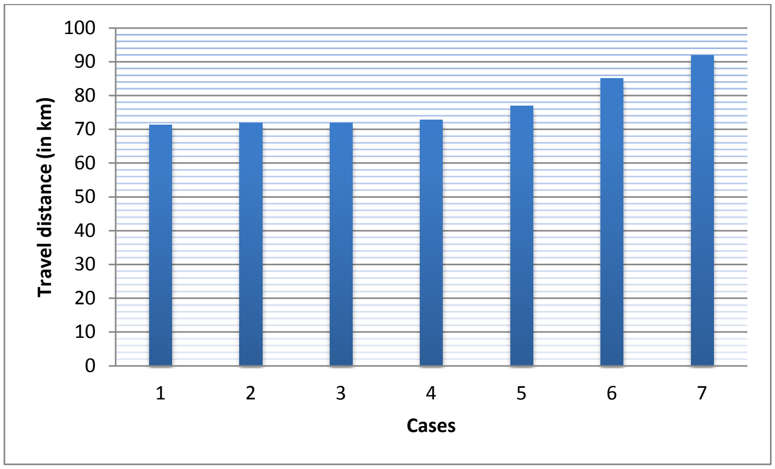


Figure 4.5: Results of various batch compositions

Our results are illustrated in Figure 4.5. The shortest total travel distance to fulfill all 7200 requests was obtained with full integration of pickups and returns, i.e., for case 1 with 71.36km travel distance. However, we also observe that a distribution of return requests over more batches does not affect the results significantly, as for case 2 (71.95km), case 3 (71.98km), and case 4 (72.72km). For practical reasons a distribution of returns over more batches might still be advantageous to allow for some flexibility for the order picker in sorting products in the picking cart. In contrast, we find significant differences in the resulting travel distance for all cases in which returns are processes separately. While in case 5 (76.88km) and case 6 (85.03km) still some pickup and return requests are performed in the same batches, in case 7 returns and pickups are entirely separated which leads to an overall travel distance of 91.94km for case 7. Hereby we can clearly show that the integration of pickup and return requests can significantly reduce travel distance. Our experiments show savings of 22.39% between the cases 1 and 7. However, the results also indicate that the precise combination of orders and returns has less impact on the total length as long as returns are processed together with the customer orders and not in separated batches.

4.4 Conclusions

In this chapter we considered the order picker routing problem in warehouses for two types of jobs to be fulfilled, since the number of products that are returned to the warehouses by customers require to reconsider classical warehouse order picking methods. We proposed a genetic algorithm to identify order picker routes by which products returns can be returned to their storage locations, while customer orders are picked as well. In numerical experiments we demonstrated the good performance of our solution approach and evaluated the potential gains in travel distance in comparison with two constructive heuristics and the gap to optimal solution for small instances. Furthermore, we explored the most suitable manner to integrate product returns in the picking batches and found that an incorporation of many returns in fewer picking batches is slightly more profitable than a distribution of returns over all picking batches. However, the most significant savings can be gained with integrated compared to separated processing of orders and returns in general, rather than with a specific mix of orders and returns in batches. Mixed batches, however, might require more processing times in practice. Future research might be necessary to explore the impact of simultaneous order and return processing on batch preparation, picking, retrieval, and sorting time.

The following chapter suggests a step further to adjust order picking operations to today's e-commerce environment. We model the batching and routing problems in an integrated form and propose a solution approach which not only accounts for product returns, but also combines two strongly interdependent warehouse operation problems.

Chapter 5

Integrated Batching and Routing

The two warehouse operation problems we studied in the previous chapters are certainly interrelated (De Koster et al., 2007). The performance of a batching policy is to a large extent dependent on the routing method that is used. Vice versa, short routes are easier to create when the batch consists of products that are stored at proximate locations in the warehouse. A simultaneous consideration of these two problems and an integrative solution approach can therefore lead to better order picking performance. For that reason this chapter is dedicated to the presentation of a linearized, mixed-integer optimization model that describes the simultaneous batch and route formation problem in a picker-to-parts system for a pool of customer orders and product returns. Our goal is to view the order picking process as a whole and identify a schedule for multiple order pickers that specifies the appropriate batching and routing sequences while minimizing total travel costs and still meeting customer deadlines.

We integrate batching and routing into one solution approach, in accordance with Tsai et al.'s (2008) proposed nested genetic algorithm. We build on this work by (1) proposing a model that is independent from the warehouse layout, (2) suited for the incorporation of customer returns, and also designed to (3) include multiple order pickers simultaneously, who attempt to meet customer order deadlines.

According to De Koster et al. (2002), the warehouses of mail order companies might process and ship approximately 7000 customer orders per day, each of which can consist of multiple items. In addition to the customer orders that need to be picked per working shift, returns arrive at the warehouse and must be handled. We show that the presented solution algorithm can find efficient batch schedules for such large-scale problems. With its help, the order picking (i.e., batching and routing) problem can be solved holistically, and the return flow can be incorporated in an efficient manner.

Search-based algorithms and especially iterated search techniques have been shown to be suited for and applicable to the solution of warehouse order picking problems (e.g., Albareda-Sambola et al., 2009; Henn et al., 2010; Theys et al., 2010). Therefore, we study in this chapter the potential of near-optimal approaches to solve large-scale problems while also facilitating implementation in practice.

The remainder of the chapter is structured as follows: We present a linearized model of the problem under study in Section 5.1. In Section 5.2 we describe the iterated local search algorithm in detail. Section 5.3 is dedicated to the introduction of our experimental design and data samples. In Section 5.4 we summarize and discuss our numerical results. We conclude in Section 5.5.

5.1 Problem definition

We consider a warehouse order picking problem with multiple pickers in a picker-to-parts system. Our objective is to find a schedule which states a sequence of batches for each order picker and the routes that a picker uses fulfill a batch. We model this problem independent from most specific warehouse characteristics including their specific layouts, only the distances between locations need to be specified. We assume that the shortest path between any pair of locations is known. Furthermore, the number of products to be returned does not affect the design of the model itself; we simply require that the warehouse possesses only one depot. The depot is the location to which all picked products get delivered by the order pickers and all returned products initially are located in the depot; therefore, all routes of pickers begin and end at the depot. The number of available order pickers is also known. An order line refers to a single, specific product to be picked or returned, possibly with multiple quantities. An item refers to a single product, so an order line consists of one or more identical items. In turn, a customer request (pick or return) can consist of multiple order lines and is allowed to be distributed over multiple batches. However, an order line cannot be split. Customer order deadlines apply to all order lines contained, whenever they occur during the shift, whereas deadlines for returned products are set to occur at the end of the shift. Each item has a known weight, and the transport capacity of a picker is limited and cannot be exceeded at any point of a route. We assume that no single order line exceeds the transport capacity of the picking device and we anticipate that the travel speed of all order pickers and the time required to pick or return an item are constant. We detail our notations for the main parameters and decision variables in Tables 5.1 and 5.2.

The set of all locations in the warehouse is denoted by $\mathcal{N} = \{0, \ldots, N\}$, where 0 represents

$\mathcal{N} = \{0, \ldots, N\}$	Set of all locations	v	Travel speed
$\mathcal{I} = \{1, \ldots, I\}$	Set of all order lines	c_{break}	Break between subsequent routes
$\mathcal{E} = \{1, \ldots, E\}$	Set of order pickers	s	Travel cost per time unit
$dist_{n_1 n_2}$	Distance between n_1 and $n_2 \in \mathcal{N}$	t_p	Pick / return time of one item
		α	Delay penalty per item and time unit
q_i^p	Quantity of i to pick	Q	Capacity of picking device
q_i^r	Quantity of i to return	H_e	Maximum number of batches of picker $e \in \mathcal{E}$
d_i	Deadline of i relative to the start of the shift	H	Total number of batches
w_i	Weight of one item in order line i		

Table 5.1: Parameter Notations

$b_i^{he} \in \{0, 1\}$	Equals 1, if i is contained in the h-th batch of picker e
$\chi_{ij}^{he} \in \{0, 1\}$	Equals 1, if location of j is visited after location of i in the h-th batch of picker e, 0 otherwise
$\psi_{ij}^{he} \in \mathbb{R}_+$	Total load of already picked items, transported along arc (i, j)
$\omega_{ij}^{he} \in \mathbb{R}_+$	Total load to be delivered, transported along this arc (i, j)
$td^{he} \in \mathbb{R}_+$	Travel distance of the route of the h-th batch of picker e
$\tau_i \in \mathbb{R}_+$	Tardiness of i
$c_i \in \mathbb{R}_+$	Completion time of i
st^{he}, $co^{he} \in \mathbb{R}_+$	Start and completion time of the h-th batch of picker e

Table 5.2: Decision Variable Notations

the depot. One request by a customer consists of either returns or orders (otherwise, the request is treated as two separate order lines), such that $q_i^r = 0$ or $q_i^p = 0$ for all $i \in \mathcal{I}$. The distance between any pair of locations $n_1, n_2 \in \mathcal{N}$ is denoted by $dist_{n_1 n_2}$. The maximum number of batches that can be assigned to order picker $e \in \mathcal{E}$ is denoted by H_e, and the total number of batches is $H = \sum_{e=1}^{E} H_e$. Because the number of batches that are built for the solution schedule is not known in advance, this construct implies that some batches might be empty. However, with our modeling approach, empty batches do not cause any travel cost and have no duration.

The decision variables χ_{ij}^{he}, ψ_{ij}^{he}, and ω_{ij}^{he} serve to model the routing decisions, comparable to the formulation provided by Mosheiov (1994). Thus, χ_{ij}^{he} is a binary variable that expresses whether the arc between the locations of order line i and j (arc (i, j)) is traveled by picker e in batch h to visit the locations i and j. In turn, ψ_{ij}^{he} denotes the load that has already been picked, and ω_{ij}^{he} refers to the load of order lines that still have to be returned in this route. Both loads are carried along the arc (i, j); the sum must not exceed the transport capacity Q.

Before we present the entire programming model, we introduce the most important interrelations between variables. The binary variable b_i^{he} describes whether order line i is contained in batch h of picker e, that is

$$b_i^{he} = \begin{cases} 1 & \text{if } i \text{ is contained in the } h\text{-th batch of } e \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I},\ h = 1, \dots, H_e,\ \forall e \in \mathcal{E}.$$

Thus, we must require that the length of the arc (i, j) between the locations of order line i and j is considered in the corresponding route only when both i and j are contained in the corresponding batch. That is,

$$\begin{aligned} \chi_{ij}^{he} &\leq b_i^{he} \quad \forall i, j \in \mathcal{I},\ h = 1, \dots, H_e,\ \forall e \in \mathcal{E}, \\ \chi_{ij}^{he} &\leq b_j^{he} \quad \forall i, j \in \mathcal{I},\ h = 1, \dots, H_e,\ \forall e \in \mathcal{E}. \end{aligned}$$

The travel distance of one batch can then be described by

$$td^{he} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}} dist_{n(i)n(j)} \cdot \chi_{ij}^{he} \quad \forall h = 1, \dots, H_e,\ \forall e \in \mathcal{E},$$

where $n(i)$ and $n(j)$ denote the locations of order lines i and j, respectively.

The start and completion times of all batches of an order picker e depend on each other and can be iteratively described by

$$\begin{aligned} st^{he} &= 0 && \text{for } h = 1, \forall e \in \mathcal{E}, \\ co^{he} &= st^{he} + \frac{td^{he}}{v} && \forall h = 1, \dots, H_e,\ \forall e \in \mathcal{E}, \\ st^{he} &= co^{(h-1)e} + c_{break} && \forall h = 2, \dots, H_e,\ \forall e \in \mathcal{E}, \end{aligned}$$

where v is the travel speed, and c_{break} is a break between two routes of a picker to prepare for the coming route or to take a break. The completion time c_i of an order line i then can be derived with the help of a technical variable $c_i^{he} \geq 0$, using

$$\begin{aligned} c_i^{he} &= co^{he} \cdot b_i^{he} \qquad \forall i \in \mathcal{I}, \forall h = 1, \ldots, H_e, \ \forall e \in \mathcal{E}, \\ c_i &= \sum_{e=1}^{E} \sum_{h=1}^{He} c_i^{he} \qquad \forall i \in \mathcal{I}, \end{aligned} \tag{5.1}$$

where c_i^{he} denotes the completion time of an order line in a particular batch and is equal to the completion time of this batch, if it is contained in it. For all other batches it is equal to 0. To maintain a linear model formulation, we transform constraint (5.1) into a linear expression by requiring

$$\begin{aligned} co^{he} - c_i^{he} &= \kappa_{ihe}^1 && \forall i \in \mathcal{I}, \forall h = 1, \ldots, H_e, \ \forall e \in \mathcal{E}, \\ c_i^{he} &= \kappa_{ihe}^2 && \forall i \in \mathcal{I}, \forall h = 1, \ldots, H_e, \ \forall e \in \mathcal{E}, \\ \kappa_{ihe}^1 &\leq (1 - b_i^{he}) \cdot A && \forall i \in \mathcal{I}, \forall h = 1, \ldots, H_e, \ \forall e \in \mathcal{E}, \\ \kappa_{ihe}^2 &\leq b_i^{he} \cdot A && \forall i \in \mathcal{I}, \forall h = 1, \ldots, H_e, \ \forall e \in \mathcal{E}, \end{aligned}$$

where $\kappa_{ihe}^1, \kappa_{ihe}^2 \geq 0$, and $A > 0$ is a sufficiently large constant.

The potential tardiness of an order line i then can be derived by

$$\tau_i = \max\{0, c_i - d_i\} \quad \forall i \in \mathcal{I}. \tag{5.2}$$

In our case, the description of the delay τ_i can be simplified by requiring the linear restrictions

$$\begin{aligned} \tau_i - (c_i - d_i) &\geq 0 \quad \forall i \in \mathcal{I}, \\ \tau_i &\geq 0 \quad \forall i \in \mathcal{I}, \end{aligned}$$

because the delay τ_i is directly minimized in the objective function.

In total, the mixed-integer linear programming model, which minimizes the total cost and tardiness, takes the following form:

$$\min \left\{ s \cdot \left(\underbrace{\frac{1}{v} \cdot \sum_{e=1}^{E} \sum_{h=1}^{H_e} \sum_{i=0}^{I} \sum_{j=0}^{I} dist_{n(i)n(j)} \cdot \chi_{ij}^{he}}_{\text{travel time}} + \underbrace{\sum_{i=1}^{I} (q_i^p + q_i^r) \cdot t_p}_{\text{pick time}} \right) + \alpha \cdot \underbrace{\sum_{i=1}^{I} \tau_i \cdot (q_i^p + q_i^r)}_{\text{delay time}} \right\} \tag{5.3}$$

subject to

$$\sum_{e=1}^{E}\sum_{h=1}^{H_e} b_i^{he} = 1 \qquad \forall i \in \mathcal{I},\ i \neq 0 \tag{5.4}$$

$$\sum_{e=1}^{E}\sum_{h=1}^{H_e}\sum_{i=0}^{I} \chi_{ij}^{he} = 1 \qquad \forall j \in \mathcal{I},\ j \neq 0 \tag{5.5}$$

$$\sum_{e=1}^{E}\sum_{h=1}^{H_e}\sum_{j=0}^{I} \chi_{ij}^{he} = 1 \qquad \forall i \in \mathcal{I},\ i \neq 0 \tag{5.6}$$

$$\sum_{j=0}^{I} \chi_{0j}^{he} \leq 1 \qquad \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.7}$$

$$\sum_{j=0}^{I} \psi_{ij}^{he} - \sum_{k=0}^{I} \psi_{ki}^{he} = \begin{cases} q_i^p \cdot w_i \cdot b_i^{he} & \text{if } i \neq 0 \\ -\sum_{l\in\mathcal{I}} q_l^p \cdot w_l \cdot b_l^{he} & \text{if } i = 0 \end{cases} \tag{5.8}$$

$$\forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E}$$

$$\sum_{j=0}^{I} \omega_{ij}^{he} - \sum_{k=0}^{I} \omega_{ki}^{he} = \begin{cases} -q_i^d \cdot w_i \cdot b_i^{he} & \text{if } i \neq 0 \\ \sum_{l\in\mathcal{I}} q_l^d \cdot w_l \cdot b_l^{he} & \text{if } i = 0 \end{cases} \tag{5.9}$$

$$\forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E}$$

$$\psi_{ij}^{he} + \omega_{ij}^{he} \leq Q\chi_{ij}^{he} \qquad \forall i,j \in \mathcal{I},\ h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.10}$$

$$\chi_{ij}^{he} \leq b_i^{he} \qquad \forall i,j \in \mathcal{I},\ h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.11}$$

$$\chi_{ij}^{he} \leq b_j^{he} \qquad \forall i,j \in \mathcal{I},\ h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.12}$$

$$st^{he} = 0 \qquad \text{for } h = 1, \forall e \in \mathcal{E} \tag{5.13}$$

$$co^{he} = st^{he} + \frac{td^{he}}{v} \qquad \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.14}$$

$$st^{he} = co^{(h-1)e} + c_{break} \qquad \forall h = 2,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.15}$$

$$c_i = \sum_{e=1}^{E}\sum_{h=1}^{He} c_i^{he} \qquad \forall i \in \mathcal{I} \tag{5.16}$$

$$co^{he} - c_i^{he} = \kappa_{ihe}^1 \qquad \forall i \in \mathcal{I}, \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.17}$$

$$c_i^{he} = \kappa_{ihe}^2 \qquad \forall i \in \mathcal{I}, \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.18}$$

$$\kappa_{ihe}^1 \leq (1 - b_i^{he}) \cdot A \qquad \forall i \in \mathcal{I}, \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.19}$$

$$\kappa_{ihe}^2 \leq b_i^{he} \cdot A \qquad \forall i \in \mathcal{I}, \forall h = 1,\dots,H_e,\ \forall e \in \mathcal{E} \tag{5.20}$$

$$\tau_i \geq c_i - d_i \qquad \forall i \in \mathcal{I} \tag{5.21}$$

$$\chi_{ij}^{he},\ b_i^{he} \in \{0,1\} \qquad \forall i,j \in \mathcal{I},\ h = 1,\dots,H_e,\ \forall e \in \mathcal{E}$$

$$\psi_{ij}^{he},\ \omega_{ij}^{he},\ \kappa_{ihe}^1,\ \kappa_{ihe}^2,\ \tau_i \geq 0 \qquad \forall i,j \in \mathcal{I},\ h = 1,\dots,H_e,\ \forall e \in \mathcal{E}$$

The objective function in Equation (5.3) includes the total travel and pick time costs of all order pickers, as well as the penalty cost α for order lines fulfilled too late. Constraint (5.4) ensures that each order line is contained in exactly one batch. With the help of the constraints in Equations (5.5) and (5.6), we also confirm that each location at which order lines need to be picked or returned gets visited exactly once in one route. Constraint (5.7) prohibits multiple visits to the depot within one route, to avoid that the capacity restriction can be eluded. With Constraints (5.8) and (5.9), we keep track of the currently transported load during each route. Equation (5.10) limits the transported load at any point of an order picker's route to the maximum transport capacity Q. Constraints (5.11) and (5.12) are required to express that an arc (i, j) can only be traveled if i and j are contained in the same batch. In Equations (5.13) - (5.15), we express the relationships of the start and completion times of the batches addressed by same order picker. Constraint (5.16) defines the completion time of an order line, and Equations (5.17) - (5.20) replace the nonlinear Equation (5.1). Finally, Equation (5.21) replaces the maximum expression in Equation (5.2) for tardiness τ_i.

Overall, we seek a solution that assigns all order lines to a schedule, consisting of a fixed number of order pickers and set a number of batches per order picker. A single order picker deals with his or her batches sequentially. Order lines are included into batches in such a way that the capacity restriction of the picking device is not exceeded at any point of any route, the deadlines of order lines are met, and the total travel distance remains as short as possible. Some batches might be empty. However, using the proposed objective function in Equation (5.3), we note that these batches do not cause any travel costs and have no duration.

To provide insights into the problem complexity, let us first consider the order picker routing problem with integrated order picking and return handling alone. For one order picker and one batch, this problem reduces to a classical Traveling Salesman Problem (TSP), if the transport capacity of the order picker is large enough (Hernández-Pérez and Salazar-González, 2004). With respect to TSP problems for warehouse routing, some previous research studies propose polynomial time solution approaches for specific layouts (Ratliff and Rosenthal, 1983; Roodbergen and De Koster, 2001b). However, Theys et al. (2010) argue that the problem turns into a hard problem again for warehouse layouts with more than three cross aisles. The assignment of deadlines to customer orders does not further complicate our problem, because deadline violations are penalized in the objective function, but they are not restricted. Thus, the decision variant of our optimization program combines several Traveling Salesman Problems, each of which with simultaneous order picking and return processing. Therefore, and considering the size of the instances to be solved in practice, we propose a meta-heuristic approach, namely an iterated local search approach, to determine near-optimal solutions within reasonable computation times. In the following section, we explain this solution approach in detail.

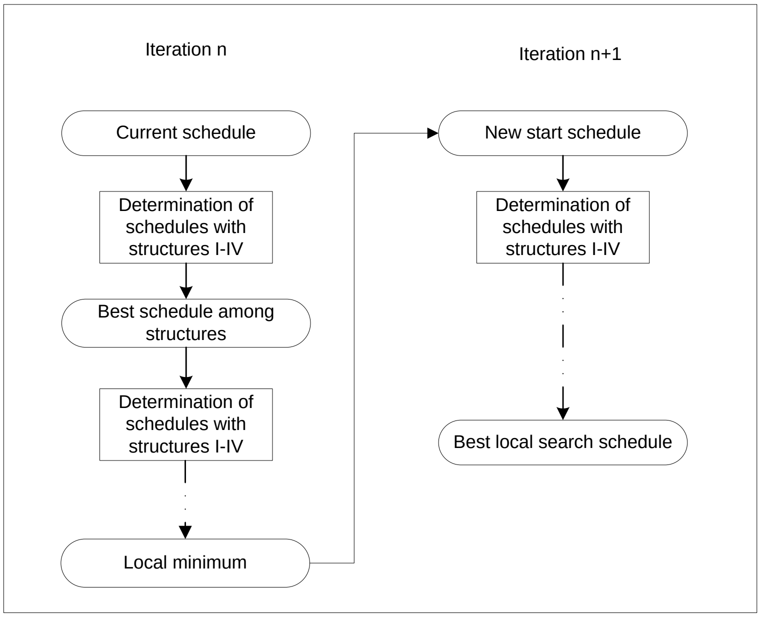

Iteration n
Iteration n+1
Current schedule
Determination of schedules with structures I-IV
Best schedule among structures
Determination of schedules with structures I-IV
Local minimum
New start schedule
Determination of schedules with structures I-IV
Best local search schedule

Figure 5.1: Algorithm design

5.2 Iterated local search

The presented approach is an iterated local search algorithm, capable of dealing with large numbers of customer orders. With the help of a local search heuristic, a local minimum can be derived from an initial solution. The iteration of this local search procedure offers a broader search perspective. In each iteration, random changes of the local minimum create a new start solution for the subsequent local search. The algorithm stops when a pre-set number of iterations is reached and the best solution found during the procedure arises. The algorithm design is illustrated in Figure 5.1.

For the initial solution, preliminary experiments with variously constructed starting points showed that the choice of the initial solution had no significant effects on the performance of the final schedule. Therefore, we selected an intuitive construction of initial solutions by including order lines iteratively in batches in the sequence in which they appear in the (unsorted) set of all order lines (data set). The first batches of all order pickers are filled before the order lines are included in the second batch of any order picker. When including order lines, we limit the maximum weight of the batches to the transport capacity, without considering the order line type (pick or return) or their deadlines.

5.2.1 Local search

Local search algorithms are often accompanied by adjustments of the neighborhood search operators during the procedure (e.g., Albareda-Sambola et al., 2009). We use four structures, each of which addresses a different attribute of a solution, which in total covers our multi faceted objective. Combining several neighborhood structures can be a helpful method to explore the search region broadly (Bianchessi and Righini, 2007; Stenger et al., 2013). From each structure, we can identify a candidate solution and select the best performing candidate solution which have been identified by the neighborhood search operators which then informs the next local search step.

In contrast with the programming model, which we derived in the previous section, capacity violations are not prohibited in the solution algorithm, but they are penalized with additional costs. We take the capacity constraint into account, without further complicating the solution approach, by setting the violation costs sufficiently high, such that no local search procedure stops before a solution without capacity violations has been derived. The four neighborhood search operators can recombine or re-sort batches. We search for improvements through the removal and reinsertion of outliers (e.g., order lines that require long detours) and order lines that violate deadlines. During each local search, we control for the focus of the algorithm by adjusting the search region and the impact of certain structures. Thus, we first find a batch formation on the basis of the proximity of locations of order lines. In subsequent local search steps, we assign more weight to refining the schedule, by removing capacity violations, tardiness, and detours. If no better solution can be found in any of the neighborhood structures, the local search stops. To measure the performance of a candidate solution, we use Equation (5.3) to define the total costs TOC, to which we add penalty costs for transport capacity violations. These violation costs are added to the objective value for each route step at which a capacity violation occurs in the routing sequence. We set the start time of the first batch of each picker to be 0. The remaining start and completion times are determined iteratively by the current travel time of each batch.

Neighborhood structure I: similarity of batches

For any pair of batches B_1 and B_2 in the current solution, we define a similarity expression by

$$B_1 \text{ and } B_2 \text{ are similar if } \left\{ \begin{array}{lcl} |st_{B_1} - st_{B_2}| & \leq & tol_{sim} \quad \text{and} \\ |co_{B_1} - co_{B_2}| & \leq & tol_{sim} \ , \end{array} \right. \tag{5.22}$$

where tol_{sim} is a positive tolerance parameter that is high in the beginning of a local search, decreases during the local search, and then resets to an initial value at the beginning of a new local search. Thus, any pair of batches is similar if they have similar start times and similar completions times according to the current tolerance value tol_{sim}. We elaborate on the specific selection of control parameters in the following section. A suitable selection of tol_{sim} depends on the instance size and the average duration of routes. At the beginning of a local search, the value of tol_{sim} should facilitate similarity among all batches with the same number (e.g., the first batch of each picker is similar to the first batch of other pickers) and the two subsequent batches of the same picker. We define a neighbored solution according to neighborhood structure I, as one obtained by the recombination of the order lines of two similar batches, B_1 and B_2. Therefore, we divide the set of order lines in $B_1 \cup B_2$ into two new batches by applying single linkage cluster merging. For this procedure we define clusters which are sets of order lines that we aim to merge to two new batches. Each cluster initially contains only one order line of the set $B_1 \cup B_2$. Iteratively, we merge two clusters $c_1,\ c_2 \subset B_1 \cup B_2$, if they fulfill the following condition:

$$(c_1, c_2) = \arg \min_{c_1,\ c_2 \subseteq \mathcal{C}} \left\{ dist_{n(i_1),n(i_2)} |\ i_1 \in c_1,\ i_2 \in c_2,\ c_1 \neq c_2 \right\},$$

where $\mathcal{C}$ denotes the set of all clusters, and $n(i)$ is the location of order line i. The cluster merging procedure stops if two clusters are left or if the total load of orders or returns reaches the transport capacity. In the latter cases, all remaining clusters are merged to a single second cluster. Single linkage cluster merging appears to be suited, because it combines two clusters, if the first cluster contains an order line that is as close as possible to an order line of the second cluster, that is we search for the minimal additional traveled distance when merging two sets of order lines. We also determine the performance when both batches merge for small batch pairs. If two entire batches merge an empty batch might appear in the schedule. In that case, all later batches of this picker shift. To obtain a new solution, we specify the sequence of locations to be visited in an S-shape routing plan, while respecting the capacity restriction. That is, the locations are visited in an S-shaped route through the warehouse and the orders that would exceed the capacity get skipped, to be picked on the order picker's way back to the depot. For warehouse layouts in which S-shape routing is not possible, the route can be sorted according to another intuitive routing policy. By decreasing the tolerance tol_{sim} successively, we allow for a broader search for recombination improvements in the beginning and reduce the computational effort in later local search steps when a rough batch schedule has been found. The best solution among all recombinations of pairs of similar batches is the candidate solution according to neighborhood structure I.

Neighborhood structure II: outlier

For each batch in the current solution, we define an outlier as an order line for which either long detours must be made, which causes capacity violations, or its removal leads to significant reductions of deadline violations for later fulfilled order lines by the same order picker. To detect these outliers, we compute the objective value for each order picker separately. We compare its performance (i.e., travel distance, cost of capacity violations, and delay costs) of this order picker with the resulting performance after one order line has been removed. The travel distance is thereby obtained by the summation of all shortest paths between two subsequent locations including the depot at the beginning and at the end of each route. If performance increases and the increase exceeds a certain value tol_{out}, the corresponding order line is treated as an outlier.

In this case, the outlier is tested for insertion in all batches with the same batch number and in all batches of the same order picker. Among those batches, we determine the ideal insertion batch for each outlier, that is, the batch for which the outlier causes the least additional cost relative to the savings gained by its removal from the original batch. The insertion batch may be empty. In the insertion batch, the outlier is included at the position in the sequence that creates the minimum additional cost. An outlier insertion is accepted if it improves the objective value. We also allow for multiple successive outlier shifts to obtain a neighbored solution and vary the maximum number of outliers that can be shifted per local search step. To reduce the computational effort, the search can be reduced to only a subset of the order pickers, which varies in each local search step. This subset is determined by dividing the set of order pickers (approximately) evenly into the desired number of order picker subsets, while in each local search step alternating only one subset is considered.

In total, a neighbored solution according to neighborhood structure II is any solution that results from a set of shifts of outliers from their origin batch to an insertion batch. The solution that results from all (or the maximum number of) outlier insertions that improve the objective value is the candidate solution in neighborhood structure II. Similar to the previous neighborhood search operator, we set the tolerance tol_{out} high in the beginning of a local search and then decrease it successively, to capture only the most disturbing outliers first. In contrast, the maximum number of outlier shifts n_{os} increases slowly, to increase the impact of the entire search operator in later local search steps. Appropriate values of tol_{out} and n_{os} depend on the instance size and the average distance between locations of order lines.

Neighborhood structure III: deadline violations

With the help of neighborhood search operator III, we account for the reduction of tardiness in the current schedule. The algorithm searches for order lines contained in a batch that is completed after the order line is due

$$\tau_{i^*} < co^{B_1} \tag{5.23}$$

where $i^* \in B_1$ is a late completed order line. In this case, the order line is tested for insertion in any earlier completed batch, such that the additional costs of the insertion batch are minimized. Again, the insertion batch can be an empty batch, in the case that fewer batches are assigned to another order picker and these batches are completed earlier. Let $\mathcal{B}$ denote the set of all batches. i^* is removed from its origin batch B_1 and inserted in batch B_2 at position k, if

$$B_2 = \operatorname*{argmin}_{B \in \mathcal{B}} \left\{ \min_k (dist_{n(j_{k-1})n(i^*)} + dist_{n(i^*)n(j_k)}) - dist_{n(j_{k-1})n(j_k)} |\, j_{k-1}, j_k \in B \right\}. \tag{5.24}$$

While searching for an insertion position, we also test whether a swap of order lines, instead of an insertion, can improve the objective. If any order line in the potential insertion batch is due after the completion time of the origin batch, the algorithm swaps this order line with the order line for which the deadline is violated and computes the resulting objective. The insertion of the order line or swapping of two order lines is accepted, if it improves the objective value. By facilitating swaps as well, we avoid the possibility that early completed batches are overloaded during the procedure, which could narrow flexibility in later local search steps. Therefore, we also prioritize swapping over insertion, if the resulting objective values are equal. Similar to the second search operator, this procedure is repeated up to n_{ds} times for other order lines with violated deadlines in each local search step. It stops when no deadline violations can be observed anymore, if no insertion of order lines in earlier batches can improve the objective value, or if a maximum number of insertions/swaps (per local search step) n_{ds} has been reached.

The solution that results from the number of insertions of order lines with violated deadlines into the corresponding best suitable prior completed batches is the candidate solution within the third neighborhood. n_{ds} is increased during a local search and reset to the initial value at the beginning of a new local search. We thus facilitate the reduction of deadline violations during the entire local search, but we increase the impact of structure III in later iterations, when a rough sorting based on the proximity of storage locations already has taken place. The selection of initial and change values for n_{ds} depends on the total number of order lines relative to the number of pickers.

Neighborhood structure IV: route performance

The fourth neighborhood search operator facilitates improvements within a batch and aims to reorganize the sequence of order lines to reduce travel costs. For each non-empty batch B, we first determine its total load of order lines to be picked l_B^p and the load to be returned l_B^r. A feasible route for batch B (i.e., a route that does not violate the transport capacity restriction) exists if

$$\begin{aligned} l_B^p &\leq Q \quad \text{and} \\ l_B^r &\leq Q. \end{aligned}$$

Otherwise the batch contains too many order lines to be picked, or too many order lines to be returned, and no feasible route can be found. Batches for which a feasible route exists initiate the application of the cheapest feasible insertion heuristic (CFI) to redesign the route (Mosheiov, 1994; Hansen and Mladenović, 2001). Starting with a route that contains only two randomly selected locations to be visited in the route, any new order line gets inserted at the position in the sequence at which it causes the minimum additional travel cost and does not violate the transport capacity restriction. For batches for which no feasible route exists, the transport capacity restriction is neglected, and the route is built only according to the cheapest insertion (CI) heuristic. The capacity violation in such batches creates high costs which facilitates that those batches are treated in future local steps by neighborhood operator I or II. A reorganization of the route of any batch by CFI or CI is accepted if it improves the objective value. The solution that results from all reorganizations that improve the objective value is the candidate solution within neighborhood structure IV.

By adjusting some of the neighborhood-specific control parameters, we facilitate a changing focus of the solution algorithm during each local search (Hansen and Mladenović, 2003; **?**). For example, in the beginning of a local search, the similarity of batches affects many batch pairs, because the tolerance value is high. This procedure is computationally intensive, but it supports a broad search for improvements across large parts of the schedule through batch recombination. Later, we restrict the similarity expression, because by then most order lines have been sorted in the batches with respect to their deadlines, and recombinations of batches only lead to improvements if the batches are conducted at similar times. Similarly, structure II aims to detect and remove the most disturbing outliers first, then focuses on refining changes later.

5.2.2 Iteration

If a local minimum has been identified, such that no improvements are possible according to any of the neighborhood structures, the local minimum is randomly altered, and a new local search starts. This alteration can be achieved by random swaps of complete batches between order pickers, swaps of partial sequences of order lines with random length between batches, and swaps of single order lines, to obtain a new schedule. This schedule serves as the start solution of the subsequent local search. The total algorithm halts after a certain number of iterations and returns the best local search minimum found during the procedure.

5.3 Experimental design

In this section we provide an overview of our experimental design by introducing the warehouse layout that we mainly use for the experiments and we present the details and characteristics of our sample data sets. We perform our experiments for the same warehouse layout as we did in Chapter 4, which is depicted again in Figure 5.2 and commonly used to study order picking performance (Gademann et al., 2001; Gong and De Koster, 2009). Again, we emphasize that our solution algorithm, as well as the programming model we described in Section 5.1 are applicable irrespective of the warehouse layout. The warehouse possesses parallel aisles with the length a_{length} and width a_{width}, along with two cross aisles at the beginning and the end of the parallel aisles, which allow order pickers to switch between the parallel aisles. Products are stored only in the parallel aisles, not in the cross aisles. With respect to the storage locations of order lines, we make no differentiation between the shelves of the left and the right side in an aisle. That is, order pickers can reach locations on the left and the right side within an aisle without traveling additional distances.

For this warehouse layout, the distance between any pair of locations $n_1, n_2 \in \mathcal{N}$ is given by

$$dist_{n_1,n_2} = \begin{cases} |x_{n_1} - x_{n_2}| & \text{if } y_{n_1} = y_{n_2} \\ a_{\text{width}} \cdot |y_{n_1} - y_{n_2}| + \min\{x_{n_1} + x_{n_2}, 2a_{\text{length}} - x_{n_1} - x_{n_2}\} & \text{if } y_{n_1} \neq y_{n_2} \end{cases}.$$

where y_n corresponds to the aisle and x_n to the location in the aisle.

We used 12 randomly generated sample data sets that serve as the order and return arrival process of one working shift (8 hours) and multiple order pickers. We are not aware of research studies that deal with integrated batching and routing, while taking product returns and order line deadlines into account. Therefore, we cannot use existing data sets that would allow for a comparison with other methods. Rather, the 12 data sets provide good insights into the perfor-

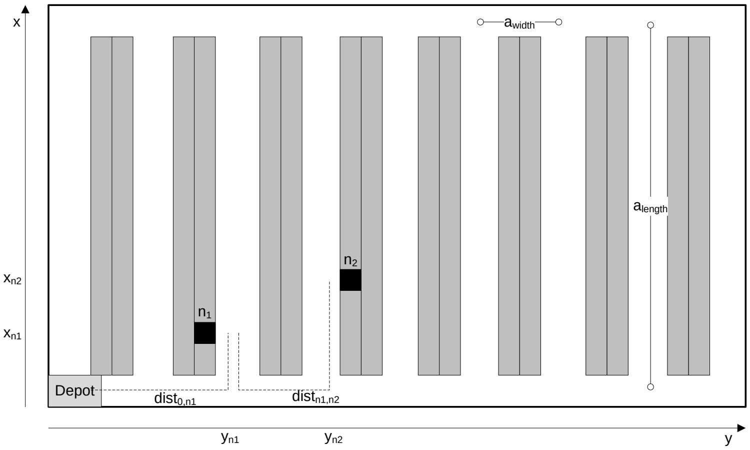


Figure 5.2: Rectangular Warehouse

mance of our solution method for all relevant characteristics of typical warehouse order picking situations. The data sets are randomly generated only by controlling for the approximate number of order lines and amount of returns in order to obtain a variety of instances. They vary in the number of order lines, the warehouse size, and the amount of returned products to fulfill as the overview of these characteristics in Table 5.3 indicates. Note that the amount of returned products is not identifiable through the return rate of a company, because the latter typically expresses the percentage of customer orders returned to the warehouse, on average. The percentages in Table 5.3, column 4, reveal the percentage of customer returns with respect to the total number of requests, such that a return amount of 30% indicates that 30 of 100 customer requests are product returns, which corresponds to a $(30/70 =)43\%$ actual return rate. Pickers move through the warehouse with a constant pace of 0.7 meters per second, and the time to pick or return one item is set to 8 seconds. In all experiments, we used travel costs of 0.54€ per minute per order picker, which equates to approximately 32€ for an order picker in each hour. Delay costs are set to 0.06€ per minute per delayed item (similar to Tsai et al., 2008). The capacity of the picking device is 80 kg for all order pickers. Finally, we set the break between two subsequent routes of the same order picker to equal $c_{break} = 5$ minutes.

The number of parallel aisles in data sets $1 - 9$ is 35, with an aisle length of 30 meters. For data sets $10 - 12$, we considered 50 aisles, with a length of 40 meters. The aisle width is set to 2.5 meters in all experiments. Deadlines for returned products are the end of the work shift (i.e., 8 hours). Deadlines for customer orders might occur during the shift, at every full hour; we assigned them randomly with a uniform distribution. The quantity of items in an order

Data set	#customers	Order lines	Return amount	Total weight (kg)	Orders (kg)	Returns (kg)
1	195	598	10%	641.0	578.1	62.9
2	227	640	20%	714.0	579.4	134.6
3	292	864	30%	904.8	528.6	376.2
4	566	1737	10%	1893.3	1764.2	129.1
5	672	2028	20%	2191.9	1766.0	425.9
6	920	2786	30%	2988.8	2097.9	890.9
7	1348	4000	10%	4299.8	3848.7	451.1
8	1706	5049	20%	5647.1	4597.3	1049.8
9	1817	5431	30%	6092.8	4099.4	1993.4
10	2029	6059	10%	6711.6	5947.0	764.6
11	2341	6995	20%	7654.3	6082.5	1571.8
12	2673	8038	30%	8897.9	6181.9	2716.0

Table 5.3: Sample Data Sets

line can be a maximum of 4, being 1 for 40% of the order lines, 2 for 30%, 3 for 20%, and 4 for 10%, resulting in a mean quantity of two items per order line. The weight of an item is accurate to 0.1 kg with a uniform distribution and a maximum of 1 kg. Similarly, the storage locations of order lines, given by the aisle number and the location in the aisle, accurate to 0.1 meters, are assigned with a uniform distribution. Random storage location assignment, as we assumed in our sample, is widespread in warehouses, because it is easy to implement, maximizes space utilization, and performs especially well when number of order lines to be fulfilled is large (Chan and Chan, 2011). However, our solution algorithm can be applied with any other storage location assignment policy. Furthermore, the number of customer requests reflects common sizes (De Koster et al., 2002). In particular, we included the smaller data sets ($1-3$) in line with previous research (e.g., Albareda-Sambola et al., 2009; Chen et al., 2005), to demonstrate the large range of application opportunities with our method. Lastly, our datasets cover typical product return rates in e-commerce businesses (Mostard et al., 2005; De Koster et al., 2002) and thus form a generally representative sample.

5.3.1 Benchmark models

Realistic instance sizes prohibit a comparison with optimal solutions. We therefore compare the performance of our method with two constructive heuristic methods. Such constructive heuristics have been widely proposed for batching (e.g., seed-order and accompanying order selection rules (Ho et al., 2008; De Koster et al., 1999)) and the routing problem (e.g., cheapest insertion heuristic); they also are often used in practice (Gademann and Van De Velde, 2005).

The first benchmark model (BM1) uses a variation of the earliest deadline first (EDF) job scheduling heuristic and organizes the route of each batch by applying the cheapest feasible insertion heuristic (CFI). In contrast with job scheduling problems, our case features no single jobs. Thus, when applying EDF, we need to make an additional decision when an order line is included in an already existing (non-empty) batch or in a new empty batch. This decision involves the consideration of the earliest deadline that occurs in the non-empty batch among already included order lines. Therefore, we consider the order lines to be sorted according to their deadlines. Note that we assigned the end of the shift to be the deadline for product returns. The first order line is included in the first batch of the first order picker. This batch is filled up with the next order lines. We determine a preliminary route by visiting the locations in the sequence in which the order lines are included in the batch. The batch is considered full if (1) the inclusion of another order line would lead to a violation of the deadline of the first included order line or (2) the total load of orders or the total load of returns would exceed the transport capacity. In these cases, subsequent order lines are included in the first batch of the next order picker in the same manner. If the first batches of all order pickers are full, order lines get included in the second batch of the first order picker. At this point, potential deadline violations must be tolerated. When all order lines are included in a batch, CFI is applied for each batch to determine the final route. To begin, only two order lines of the batch form an initial route that starts and ends at the depot. Any next order line is included at the position in the sequence at which it causes the least additional travel cost and does not violate the capacity restriction.

The second benchmark model (BM2) is a constructive insertion heuristic. Again, we consider the order lines to be sorted by their deadlines. In this sequence they are included in batches. We call a batch open if it already contains order lines and is not full with respect to the transport capacity (for orders or returns). For each order line, we allow insertion in open batches only. We first include an order line in the first batch of the first order picker and denote the first batch of the second picker to be open. The subsequent order lines are tested on each position of every open batch for insertion with respect to the resulting objective value, and then they are inserted at the most cost-efficient position. The objective function corresponds to the objective we presented in Equation (5.3), with additional costs for capacity violations (similar to Ribeiro and Laporte (2012)). The high penalties for capacity violations enforce their inclusion at positions that do not lead to infeasible solutions. If an order line is included in an empty batch, a new empty batch is opened unless a maximum number of open batches has been reached. In our experiments we set the maximum number of open batches to equal the number of available order pickers. A batch is closed if the total load of orders or the total load of returns reaches the transport capacity.

5.3.2 Control parameters

The number of iterations (i.e., the number of local searches) is set to 30 for data sets $1-9$ and to 15 for data sets $10-12$, because these selections result in good solutions and reasonable computation times. To choose the remaining control parameters, we conducted preliminary tests and observed good performance with the following values. The majority of control parameters remains constant for all experiments. A change of control parameters, as described in the solution algorithm in Section 5.2, occurs in every fifth local search step. The initial time tolerance to define which batch pairs are considered similar, used in neighborhood search operator I, is $tol_{sim} = 2000\ s$, which approximately corresponds to the time that an order picker's tour takes. tol_{sim} decreases to 95% until a lower bound of 500 s is reached. The number of outliers that can be shifted by neighborhood search operator II is initially set to 20, increases by 1, and is limited to 35, which facilitates that long detours are already removed in the beginning the search operator II retains sufficient impact in later local search steps. The tolerance that defines an outlier is $tol_{out} = 1$. It decreases to 80% and is limited to 0.2. The number of shifts that can be made by neighborhood search operator III is initially set to 20 for instances with less than 5000 order lines and 40 otherwise. It increases by 1 and is limited to 40 and 60, respectively. The costs for capacity violations are set to $10€/kg$, which suffices to avoid capacity violations in all final local search solutions. To avoid unnecessary searching effort, we split the number of order pickers for neighborhood search operator II, which only searches for improvements in one-third of the order pickers per local search step. In the two subsequent local search steps, the neighbor search is applied for the remaining two-thirds of order pickers. The number of random swaps of batches and order line sequences, conducted at the end of a local search to derive a new start schedule, needs to be adjusted to the size of the data set. This step facilitates a broad view of the search region, which requires a degree of randomness that is radical enough; it also must allow for an improvement effect over several iterations. We selected the number of swaps by defining a parameter swaps

$$\text{swaps} = max\left(3, \frac{\text{\#order lines}}{120}\right).$$

The new start schedule is derived by the number of entire batch swaps, the number of sequence swaps of random length, and the number of swaps of single order lines. These values are selected by

$$\begin{aligned}
\#\text{batch swaps} &\approx \frac{\text{swaps}}{4} \\
\#\text{sequence swaps} &\approx \frac{\text{swaps}}{2} \\
\#\text{order line swaps} &\approx \text{swaps}.
\end{aligned}$$

Preliminary experiments, which we do not report here, showed that small variations of these parameters do not significantly affect the quality of solutions.

5.4 Numerical results

We demonstrate the performance of our proposed algorithm in several steps. First, the two previously described constructive heuristics, BM1 and BM2, serve as benchmarks to verify the performance of our solution approach. Second, to show the effect of a joint optimization of order batching and order picker routing, we provide a comparison with solutions that we obtained through our algorithm when implementing a fixed routing policy, instead of an iterated route optimization. Third, we illustrate the potential gains that can be achieved by integrating customer order picking and product returns in a joint process. To do so, we solve each sample data set again and in two computations, one to process orders and one to process returns. We compare the results with those of integrated processing. Fourth, to demonstrate the generalizability of the solution algorithm, we perform experimental tests using a warehouse layout with an additional middle cross aisle. The results are summarized in Table 5.4.

5.4.1 Comparison with benchmark models

To solve the sample data sets with the two previously described heuristics, we observed that the solutions - especially of BM1 - are sensitive to the number of available order pickers. Insufficient order pickers cause enormous deadline violation costs, because the assignment of order lines into batches is relatively rigid, and no swapping or shifting of order lines is possible later when using BM1. We solved each data set with several calculations for $2 - 25$ order pickers and used only three reasonable results of BM1 (per data set) for a comparison with our method and BM2. The results of these experiments are provided in Table 5.4, column 3 (BM1), 4 (BM2), and 5 (our iterated local search approach).

As the results show, our approach significantly outperforms both BM1 and BM2 for all datasets and all picker numbers. Relative to BM1, we achieved a cost reduction of 52.6% (standard deviation 7.72%) on average over the first 27 instances. For the largest data sets $10 - 12$

1	2	3	4	5	6	7	8
Set	# Pickers	BM1	BM2	ILSA	Fixed Routes	No Integration	ILSA Cross Aisle Layout
1	3	166.08	122.12	98.87	105.93	112.91	96.67
	4	163.23	110.12	93.72	107.23	104.63	85.82
	5	174.20	111.24	91.77	104.64	102.01	87.68
2	3	175.33	130.82	98.83	123.94	123.88	94.50
	4	177.06	116.28	93.84	115.57	117.99	91.82
	5	172.68	115.26	98.20	109.72	114.79	88.99
3	4	256.18	166.87	98.17	123.36	147.05	100.73
	5	207.06	163.49	98.88	102.90	137.06	97.17
	8	267.05	126.74	102.60	120.19	162.66	100.36
4	7	478.65	325.82	253.49	329.91	276.44	245.96
	8	442.43	301.85	249.85	331.81	280.26	239.83
	9	432.77	305.40	251.97	329.49	274.93	239.56
5	8	507.23	352.96	270.54	333.21	316.50	256.62
	9	504.92	347.04	273.35	336.28	327.74	257.24
	10	515.47	319.80	258.44	321.32	319.76	242.83
6	8	806.85	510.11	343.78	410.16	411.83	328.39
	9	802.13	499.04	336.45	412.35	413.19	316.40
	10	705.79	484.07	333.18	422.92	420.03	317.37
7	12	1560.10	699.49	538.71	622.80	588.28	526.80
	13	1263.72	683.63	536.29	639.64	593.54	523.90
	14	1215.67	667.92	524.26	666.14	589.18	509.56
8	14	1689.77	894.95	660.11	757.18	838.50	623.57
	15	1667.47	872.77	648.94	745.12	787.09	635.62
	16	1502.21	860.55	660.41	757.73	776.47	628.00
9	14	1694.76	985.53	645.93	735.58	787.93	624.74
	15	1598.96	958.22	651.47	749.78	800.22	626.18
	16	1578.38	942.26	643.27	749.34	791.00	631.50
10	20	5874.19	1255.08	978.28	1191.99	1126.07	933.13
	21	5518.00	1238.72	1000.55	1257.34	1031.17	927.52
	22	4828.81	1197.41	976.95	1239.14	1086.93	922.46
11	20	6438.19	1477.99	1090.11	——-	1230.30	1012.68
	21	6129.53	1454.73	1092.31	1277.89	1230.37	1024.67
	22	5641.09	1427.63	1060.38	1379.76	1224.75	992.38
12	20	6677.17	1750.86	1191.35	——-	1397.23	1121.50
	21	5483.92	1740.34	1193.94	——-	1392.84	1109.76
	22	5266.06	1701.28	1169.86	1381.29	1356.25	1098.16

Table 5.4: Overview of the total costs (in €) for various experiments

BM1 cannot offer reasonable results with realistic numbers of order pickers. Compared with BM2, the total costs are on average 24.9% (standard deviation 6.76%) lower. In each experiment our solution approach derived results without capacity violation costs. Deadline violation costs were observed in 3 of the 36 solutions in column 5 (data set 4 with 7 pickers, 8 with 14 pickers, and 11 with 20 pickers,), but they accounted for less than 1€ in each case.

The reason for this performance is that constructive heuristics suffer from an inflexibility that our search algorithm eludes. Constructive heuristics cannot make use of a global perspective when constructing solutions successively. Order lines that are included in specific batches cannot be removed later, even if that would be profitable to distribute later considered order lines. In contrast, our algorithm searches for promising candidate solution by considering all order lines and batches at any time. Even the integrative heuristic (BM2), which accounts for route and batch optimization simultaneously, builds the schedule successively. For example, switches of order lines along batches are not possible, which might lead to unsuitable assignments of later inserted order lines. Our search algorithm instead improves the initial schedule. During the entire procedure, it uses information about the full data set in its assignments, and therefore, it can make decisions from a broader perspective. Furthermore, iteration helps improve the solution by multiple local search rounds and also facilitates an improvement for the local search.

5.4.2 Joint vs. separated batch and route optimization

The comparison of the two constructive heuristics provides interesting insights into the potential gains that can be achieved from optimizing batching and routing simultaneously. Whereas BM1 (Table 5.4, column 3) initially assigns orders and returns into batches, and sorts the routing sequence in a second step, BM2 (Table 5.4, column 4) aims to find the best suitable batch for an order line while simultaneously finding its best suitable position in the routing sequence. In turn, BM2 performs on average 37.1% (standard deviation 8.26%) better than BM1 (average over data sets 1-9 only).

Table 5.4, column 6 shows the results when we implemented S-shape routing in our solution algorithm, instead of iterative route optimization. Using S-shape routing, the order picker traverses each aisle that contains pick or return locations entirely. Every second aisle is traversed in reverse direction (De Koster et al., 1999; Henn et al., 2010). Potentially skipped locations (due to the capacity constraint) are visited again in a second round in the same manner. These experiments showed similar results. The iterated local search approach (Table 5.4, column 5) achieved on average 16.5% (standard deviation 4.89%) lower costs than those accrued using an S-shape routing strategy and the same algorithm design (Table 5.4, column 6). The fixed routing variant was solved with 30 iterations for all data sets, but for three of the large experiments it did not return results with reasonable deadline violation costs. These effects can be explained partially by the general drawback of rigid routing heuristics; the distance traveled between any pair of locations is not always the shortest possible path from one location to the other, if pickers follow a fixed routing policy. In other words, the same set (and sequence) of locations to be visited might lead to a longer travel distance for any fixed routing policy compared with our approach, in which the shortest path between any pair of locations always is traveled. However, with the help of neighborhood search operator IV, which is neglected in the fixed routing variant, we can test routes in each local search step for improvement. The random selection of start points for the cheapest (feasible) insertion facilitates a test of multiple options to create a route for the same batch. Furthermore, it supports the detection of outliers, which can be removed in the following steps by neighborhood search operator II. Neighborhood search operators II and III also lose impact in the fixed routing variant. Instead of testing several potential insertion positions in one batch, fixed routing usually allows only one position for inserting an order line, which then limits the search region of the fixed routing algorithm. Overall, these results support our arguments about the high relevance of integrated batching and routing models.

5.4.3 Integrating product returns

For the purpose of demonstrating the potential gains of the integration of customer orders with product returns in the order picking process, we conducted experiments with orders and returns processed separately. We determined a schedule that fulfills all orders first. Thereafter, we reassigned the start times of the first batches of each picker to be the completion time of the corresponding last batch in the order schedule ($+c_{\text{break}}$) and repeated the calculation for returns only. The results in Table 5.4, column 7, express the total cost of orders and returns.

The total cost when returns are processed together with order picking are 15.6% (standard deviation 6.80%) lower on average, than separated processing. Separated according to the different characteristics of the data sets we find 9.6% (standard deviation 2.57%) savings for data sets with a return amount of 10% (data sets 1, 4, 7, and 10), average savings of 16.3% (standard deviation 3.49%) for data sets with a return amount of 20% (data sets 2, 5, 8, and 11), and average savings of 21.0% (standard deviation 7.59%) for data sets with a return amount of 30% (data sets 3, 6, 9, and 12). These results clearly demonstrate the potential gains to be obtained from integrating orders and returns in one joint operation.

5.4.4 Warehouses with a cross aisle

As mentioned previously, our approach is suitable for any kind of warehouse layout, in that it uses a network representation of the locations to be visited (i.e., a distance matrix). We conducted another set of experiments for the data sample $1-12$ for the situation of a two-block warehouse. That is, we assumed another cross aisle in the middle of the parallel aisles, in which order pickers can switch between parallel aisles. The width of the cross aisle determines the distance between the first and the second block and is set to $a_{\text{cross}} = 3.5$ meters. Because neighborhood search operator I determines a preliminary routing sequence by S-shape routing, we adjusted it for these experiments. Here, S-shape routing is applied for the first block only, and thereafter for the second block. Potentially skipped locations, due to capacity problems, are dealt with during a second S-shape round in the reverse direction.

The results of these experiments are in Table 5.4, column 8. In line with previous research (Roodbergen and De Koster, 2001a; Vaughan and Petersen, 1999), we find that an additional cross aisle is advantageous because it provides more flexibility for building good routes. In comparison with the previously introduced warehouse layout depicted in Figure 5.2, we find on average 4.5% (standard deviation 2.30%) lower total costs if an additional cross aisle is available.

5.4.5 Computation times

All methods were programmed in C++, and we conducted the computational experiments on an Intel(R) Core(TM) i7 (3.40GHz) processor.

For the first benchmark model BM1, all solutions were obtained in a few seconds. For the second benchmark model, computation times varied between approximately 2 minutes (data set 1) and 2.5 hours (data set 12). With our solution algorithm, we solved instances $1-9$ by applying 30 iterations (for all experiments in columns $5-8$). Computation times largely depended on the size of the data set and varied from 1 min (data set 1) to 4 hours (data set 9). Data sets $10-12$ were solved with 15 iterations and took at maximum 7 hours (data set 12, 22 pickers). Differences in computation times between the two warehouse layouts were not observed. With the fixed routing variant, the results were computed slightly faster than with the algorithm that also optimizes the route.

The complexity that results from the integration of two optimization problems, together with the large size of our data sets, leads to higher computation times than when using more intuitive methods, such as BM1 and BM2. Yet good solutions and large improvements compared with the two benchmark heuristics emerged already from the early iterations for all

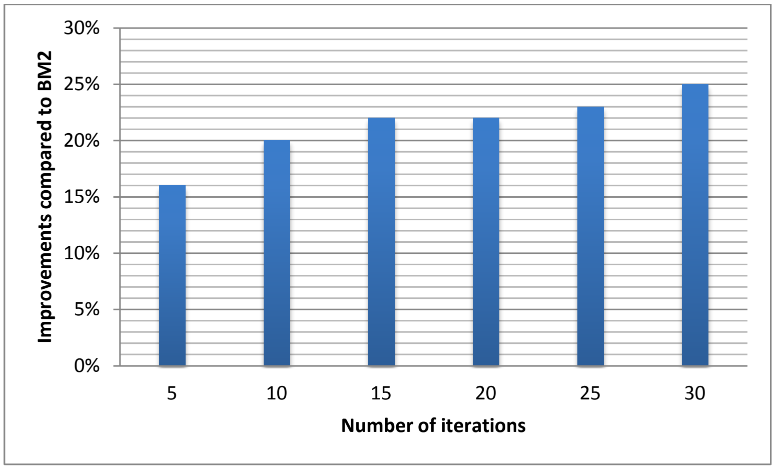


Figure 5.3: Improvements during iterations relative to the results of BM2

instances. Figure 5.3 provides an overview of the average improvements compared with BM2 for specified numbers of iterations. For up to 15 iterations, these averages cover all data sets, whereas for more iterations, the averages were determined for data sets $1-9$ only.

As Figure 5.3 shows, our solution approach outperforms the comparison model, even with just five iterations. Yet significant improvements can be obtained, if longer computation times can be facilitated and the algorithm can make the best use of the improvement effect that accompanies alternations to a current local search minimum. The resulting solution schedule organizes the order picking activities for an entire working shift and all available order pickers at one time. Thus, we consider computation time limits less restrictive because the schedule for the coming working shift in the warehouse can be calculated during the current shift. Nevertheless, Figure 5.3 also illustrates that with shorter computation times good results can be achieved.

As indicated in the Section 5.3, neighborhood search operator II is only applied for one-third of the order pickers per local search steps. A further reduction in the number of order pickers affected by structure II during each local search step could accelerate the procedure. Similarly, this division could be incorporated when searching in neighborhood III.

We studied the option to apply all neighborhood search operators subsequently, instead of accepting only the best solution among all structures in each local search step. Yet with this algorithm version, we suffered significant quality losses in the resulting solution (up to 30%). A subsequent application of the structures obviously reduces the required number of local search steps until a local minimum has been identified. However, this quick convergence of the local searches provides fewer opportunities to find good local optima and thereby prohibits a broad exploration of the search region, such that it exerts ultimately a negative effect on overall performance.

5.5 Conclusions

In this chapter we studied the combination of two highly relevant warehouse operation problems, namely, batching and order picker routing. Both are essential for the overall warehouse performance, because order picking accounts for the major part of the warehouse operation costs. The numerical experiments reported in this chapter show that there is a clear potential to save costs if the two interdependent processes order batching and order picker routing are considered simultaneously. The integration yields 16.5% savings in the total cost. We proposed an ILP formulation that combines the two problems, accounts for customer order deadlines, and can incorporate potential product returns in the order picking process as well. Our experiments also reveal that small return amounts can almost completely be processed without additional labor effort by integration with the picking. Furthermore, our proposed model is applicable for multiple order pickers, which is especially of interest for batching problems in which orders are accompanied by deadlines. We have presented a solution algorithm that can also solve large problem instances. We observed potentially significant cost reductions compared with two constructive heuristic methods (52.6% compared with BM1 and 24.9% compared with BM2), as well as a search-based method, in which the routing method is not optimized but instead remains fixed.

In summary, our model and solution algorithm help to organize the order picking process as a whole; to account for the interdependencies between batching and routing, multiple pickers, and deadlines; and to tackle increasing volumes of product returns.

Concluding remarks on Chapter 3-5

These first chapters of the thesis dealt with order picking operations in warehouses and studied opportunities for integrating the forward and return flow processing for companies with many product returns. While some operations (e.g., sorting and (un-)packaging) have to remain separated for orders and returns, batching and picker routing are two warehouse operations which allow for such integration. In particular for these order picking operations a consideration of simultaneous processing of product returns and orders appears to be a promising idea to enhance performance. Order picking batches, which are obviously restricted by the transport capacity of the picker or the picking device, can be much larger than for separated order and return handling if only the batch consists of no more orders and no more returns than capacity allows. We showed that by conducting larger batches travel costs can be reduced significantly.

We dealt with the order picking problem once for isolated batch and route formation (Chapters 3 and 4) and in a second step we dealt with the problems in an integrated manner. There are good reasons which motivate these two designs: On the one hand, in practice not all warehouses have the freedom to select batching and routing policies arbitrarily. The case of the library warehouse in Chapter 3, for example, has, due to the warehouse layout, limited routing options, so that efficiency of the order picking can be increased only by focusing on the batching policy. Vice versa, short deadlines and many customer orders might restrict the batching policy to a time window based approach (see section 2.1) and leave space for improvement only with respect to the routing policy. Although we showed in Chapter 5 that an integration of batching and routing can certainly be useful, isolated approaches are therefore important as well.

The integrated approach in Chapter 5 extends the application area by approaching the problem for multiple order pickers simultaneously. Especially in e-commerce deadlines to process customer orders are given and can be very short. In this case it is important to consider all available order pickers at the time to decide which picker conducts which tasks in which sequence. Therefore a holistic approach to schedule an order picking shift is required which implies a simultaneous batch and route formation and we presented an appropriate solution approach for large-scale problems in Chapter 5.

Chapter 6

Staff Planning in Warehouses under Uncertainty

This and the following chapter deal with another major challenge that e-commerce warehouses are confronted with. The opportunity for customers to order online at home and at any time, the high competition, and the need for short delivery times have resulted in high planning uncertainties related to several warehousing issues. Referring to the classification of Rouwenhorst et al. (2000) on warehouse problems, especially operational level decision problems are affected by uncertainties with internal and external sources (Gong and De Koster, 2011), due to the little space for failures or delay on the one hand and tight budgets on the other.

Especially decisions related to staff scheduling, such as the assignment of full time staff to specific tasks and working shifts, the allocation of flexible labor sources (e.g., hourly employees), or the utilization of external workforce via temporary employment agencies, shape essential decision problems on which uncertainties can have a major impact. This applies to warehouses as well as to many other supply chain links. Labor is often a large cost component (Eveborn and Rönnqvist, 2004) and must therefore be used most efficiently. In warehouses, delays of order fulfillment are hardly an option and the time windows to pick an order and to prepare it for shipment have become very short (Gunasekaran et al., 2001). On the other hand, over-staffing needs a careful consideration to obey budget restrictions due to the costs involved.

Chapter 7 proposes a way to control risks in warehouse staff scheduling problems by means of appropriate risk optimization modeling approaches. It contains a risk model analysis and, based on that, derives a decision support tool to guide warehouse managers in making risk averse staff scheduling decisions.

Chapter 7

A Decision Tool for Risk Control in Warehouse Staff Planning

Staff planning plays a vital role in the economic success in many industries. Labor is typically accompanied with high costs in many production, retailing, and service environments (Ernst et al., 2004). Excellent staff scheduling policies become crucial when fluctuating workforce demand or other uncertainties are present (Gong and De Koster, 2011), while shortage of labor might have a serious impact on performance and causes bull whip effects in an entire supply chain. Furthermore, diverse working contracts, specific skills and tasks, and the variance in productivity of employees might complicate staffing decisions (Fowler et al., 2008). Nevertheless, due to the high costs and risks involved, staffing decisions must be made as accurately as possible by incorporating all available information.

Our aim in this chapter is to consider warehouse staffing problems and to propose a decision support tool which assists warehouse managers to make the right choice of risk control in order to identify suitable staffing policies that match their goals. In commercial warehouses products have to be picked, packed, and prepared for shipping in response to customer orders. Thereby, typically large numbers of small orders have to be fulfilled, which involve higher uncertainties than the shipping of large order quantities for in-store buying. In contrast to stores that order on a regular basis, the consumers' online shopping behavior is less easy to predict. Short response times to customer orders and accuracy in delivery times are essential service performance indicators (Keeney, 1999). Besides demand fluctuations also absenteeism of workers and the amount of returns to be processed are examples of the stochastic influences that affect labor requirements in warehouses. Zhang et al. (2009), for example, reported a dramatic increase of costs caused by small variations in the amount of absent workers, which suggests to also prepare for such events to a certain extent. Demand peaks have to be taken into account and especially distribution centers that perform the fulfillment for other companies are often informed about special promotions, advertisements, or discount offers of their clients on very short notice. The best possible scheduling for the upcoming planning period is essential, while accounting for uncertainties becomes inevitable if the impact of even small failures is high. Any inaccuracy can lead to lost sales or unnecessarily high labor costs when external personnel needs to be hired on short-notice, and mismanagement can even lead to a loss of the client or the competitive position.

Since staffing decisions are very sensitive to the given problem, they are usually approached for specific application areas in the literature (Ernst et al., 2004). The staffing in e-commerce warehouses differs in two ways from other application areas. First, the demand fluctuation accompanied by Internet retailing is high (Gong and De Koster, 2011) due to the flexibility for customers to shop independent from opening hours (Pechtl, 2003). Furthermore, high product return rates are a result of many online purchases; their number and the labor effort involved creates additional uncertainty for the labor demand. Second, in most settings warehouse workers do not require specific skills to perform one or the other task. In contrast to other staff scheduling problems, here it remains the problem of scheduling the number of working hours for each employee to fulfill a specific workload.

Staff scheduling problems with uncertain labor demand have been approached in previous literature (Bard et al., 2007; Liao et al., 2012; Jeang, 1994). However, variating demand patterns in e-commerce warehouses motivate a further exploration of risk aversion tools. For example, the common approach of replacing the stochastic parameters with their deterministic expected values will only lead to well performing staffing policies as long as the realizations of the stochastic parameters correspond approximately with the expectation. Especially in the context of warehouse operations staff predications are only reliable to a certain extent and variations are high. In this case, staffing policies which prepare the ”mean case” might fail to provide reasonable performance for the majority of possible scenarios. Multistage stochastic models are a more sophisticated approach to tackle the uncertainties in staffing problems. Bard et al. (2007) demonstrate in their article on staff scheduling in mail processing potentially 4% lower costs when recourse decisions are allowed during the planning horizon compared to the outcomes of the problem that solely optimizes expectations of the entire time horizon.

To develop the tool we examine five optimization approaches to deal with uncertainties in warehouse staff planning. We study the behavior of various risk models, which are often used in financial risk management, in a representative warehouse situation for a variety of de-

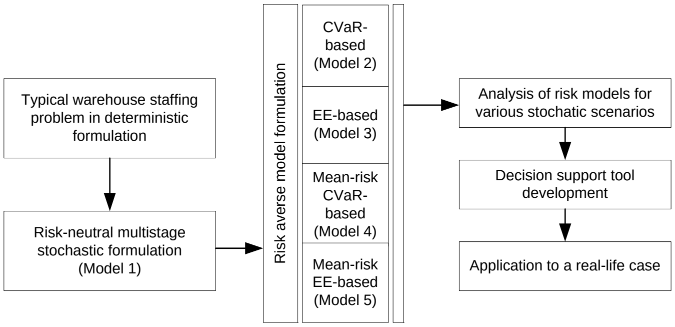

Typical warehouse staffing problem in deterministic formulation
Risk-neutral multistage stochastic formulation (Model 1)
Risk averse model formulation
CVaR-based (Model 2)
EE-based (Model 3)
Mean-risk CVaR-based (Model 4)
Mean-risk EE-based (Model 5)
Analysis of risk models for various stochatic scenarios
Decision support tool development
Application to a real-life case

Figure 7.1: Research design

mand and shortage scenarios. The first model is a classical multistage stochastic programming approach with the objective to minimize the expected total costs. Two other models utilize multistage risk measures, namely the multi-period conditional value at risk (CVaR) and the multi-period expected excess (EE), as the objective function to be minimized. Lastly, we incorporate two so-called mean-risk modeling approaches, each of which being based on one of the risk measures CVaR and EE.

The aim of this chapter is to extend traditional stochastic programming methods by analyzing the potential of risk-averse optimization strategies which rely on the CVaR and the EE for warehouse staffing problems. We propose a decision support tool that guides warehouse managers in their choice of risk control strategies in order to identify staffing policies that match their purposes. Our research design is sketched in Figure 7.1. After explaining the five models we introduce the set of warehouse scenarios for which the models are studied. The tool is designed with the help of numerical experiments for a warehouse staffing problem and eventually tested with the help of a real-case example of a Dutch commercial warehouse.

The remainder of this chapter is organized as follows. We give an overview of related literature in Section 7.1. Thereafter, Section 7.2 introduces the warehouse staffing problem we are dealing with in a deterministic version first. In Section 7.3 we specify the five multistage stochastic models on which the decision support tool is based and we provide an overview of our experimental design in Section 7.4. The development of the decision support tool on the basis of the numerical outcomes is given in Section 7.5. Section 7.6 is dedicated to a case application to test the tool. We conclude this chapter in Section 7.7.

7.1 Literature

As labor typically contributes highly to the overall cost of a company, the need for accurate models to determine staff levels, schedules, and training policies is evident. Staff scheduling problems are therefore well-studied in the literature. For an extensive review on rostering and personnel planning models and methods we refer to Ernst et al. (2004).

Typically, staffing problems are studied for specific application areas due to the uniqueness and diversity of the underlying optimization problems (Ernst et al., 2004). Many solution methods can be found for the nursing sector (e.g., Jeang, 1994; Eveborn et al., 2006), which implies, for example, a careful consideration of people's skills, or potentially preferred staff members for certain patients. Others focus on call center agencies (Aksin et al., 2007), where the response rates to calls have to be very short and the staffing problem is restricted by a certain service level constraint (Roubos, 2012). Airline crew scheduling adds complexity through geographical factors to the problem (e.g., Barnhart et al., 2003; Schaefer et al., 2005; Dunbar et al., 2012). In turn, multi-department problems are staffing problems in which several different types of work have to be fulfilled and not each employee is capable of performing one or the other type of work or the productivity for different departments varies (e.g., Brusco, 2008; Fowler et al., 2008). Also in the service industry staffing problems can be found, where labor shortage leads to lower attention to the customer's needs and thereby to lower perceived service quality (Díaz et al., 2002). A commonality among all these staffing problems is the uncertainty in labor demands, while the problem constraints, and thereby problem complexity and suitability of specific solution approaches, vary for different application areas.

Stochastic models are most prominent for staffing problems. Abernathy et al. (1973) propose a stochastic model in the nursing context and an iterative solution method to maximize workforce utility. Other approaches deal with less constrained problems and pay more attention to uncertain demands. Sadjadi et al. (2011) consider a staffing problem for various time horizons, divided into periods. Their decision variables describe the available workforce per period which is regulated by hiring and layoff policies. The author's objective is to minimize all related costs accompanied with hiring, layoff, workforce shortage, and surplus. A genetic algorithm is proposed to solve the problem near-optimal and stochasticity is taken into account by using expectation expressions. Zhang et al. (2009) study a complex staff and equipment scheduling ILP in a mail processing context, solved by decomposition and LP relaxation. The authors observe significant cost increases caused by even small variations in the amount of absent workers.

Bard et al. (2007) consider a staffing problem in the mail processing context and derive a two-stage stochastic problem with recourse. First-stage decisions thereby concern full time labor allocation and part time workforce amount. Second-stage decisions consist of the specific assignment of part time workers to the schedule, potential shortages are recovered with flexible workforce. The authors solve the corresponding high dimensional ILP heuristically based on LP relaxation, which yields near-optimal results in acceptable computation times. Comparisons with outcomes of the expected-value problem as well as with the perfect information wait-and-see approach clearly motivate stochastic models for staffing problems; their recourse problem has led to 4% lower costs compared to the simple deterministic problem, in which uncertain parameters are replaced by constant expectations.

Liao et al. (2012) consider a stochastic staffing problem for call centers with two types of tasks. Their objective is to derive a constant staffing level throughout the planning horizon. Additional uncertainty is added to their problem by random mean arrival times of inbound calls within different periods. Similar to Bard et al. (2007), the authors highlight the necessity for sophisticated approaches for situations in which the system is very sensitive toward data variation. They propose a classical stochastic programming approach for situations with known exact distributions and a robust optimization approach to protect against uncertainties. Also Van Landeghem and Vanmaele (2002) propose a robust planning model for supply chain management. They include uncertainties that might lead to bull whip effects or unnecessary re-planning cycles. The model is based on incorporating stochastic behavior via distributions and solving the resulting simulation model with the Monte Carlo method. By means of a case application they show that re-planning effort as well as disturbances can be reduced by studying various outcomes of the stochastic parameters before a final decision is made.

Fragnière et al. (2010) dealt with an annual staff planning problem in the banking sector in which also capacity is an uncertainty source. This extension is motivated by the number of external changes, the complexity of paperwork, and other tasks in the banking industry which might create capacity uncertainties of less experienced personnel if requests requires actions varying from the standards. The authors report that expending significant cost for reserving qualified personnel solely to protect against potential operational risks is at the moment perceived to be not viable in practice. However, by means of an analysis of the shadow costs they could clearly demonstrate that a surplus of qualified personnel not only reduces the operational risks but also the overall costs.

In conclusion, we observe that previous research clearly suggests to focus on sophisticated stochastic modeling approaches for staff scheduling problems (Bard et al., 2007; Liao et al., 2012; Fragnière et al., 2010). Also the focus on a specific context, namely e-commerce warehouses in our case, is motivated by the uniqueness of each application area. The distinctiveness of warehouse staffing problems lies in the high fluctuations of the amount of placed orders, the amount of product returns, and thereby the amount of required labor to realize the short response times that are typical for online retailing situations. Risk management approaches for warehouse staffing problems are therefore of high interest. However, to the best of our knowledge robust optimization is so far the sole main approach to incorporate risk control in staffing decisions. Our research thereby extends previous literature by performing an analysis of risk modeling approaches for warehouse staffing problems and by providing a decision support tool which on the one hand involves several risk control options that differ in applicability and outcomes, and which on the other hand provides guidance for warehouse managers to select an advantageous risk control policy for their specific situation.

7.2 Problem definition

The warehouse staffing problem which we use in our experiments describes a typical situation as it occurs in commercial warehouses. A large amount of the daily work is performed by full time employees, who are hired by the company, working 40 hours per week in varying shift schedules, and the warehouse pays constant salaries to them. The maximum number of hours per day that a full time worker may work is limited, as well as the minimum number of days that a person has to be off within the planning period (Eveborn and Rönnqvist, 2004). Second, there is more flexible workforce available, which we refer to as part time staff. For part time staff we do not differentiate between single employees. Instead we incorporate a variable denoting the number of workload hours which is performed by part time staff. Third, any labor shortage to fulfill demand when full and part time staff have already been assigned has to be recovered by external workforce. Warehouses often work together with temporary employment agencies. In that way they can request workforce on short notice to fulfill exceptionally high demands due to, for example, seasonal peaks or high labor effort due to discount sales. However, such recourse decisions (i.e., for example, to hire externals) might also have different backgrounds. They might correspond to overtime performed by full time or part time staff and which is more expensive than regular working time. Also, recourse decisions can refer to the postponement of work to the next shift, which in turn can cause expensive delays in the order fulfillment. Similar to Bard et al. (2007), recourse costs exceed the regular working hour costs so that there is an incentive to find schedules with low recourse costs (i.e., low use of externals, little overtime or fewer delays). We assume all workers to have an equal productivity. In practice, there are different tasks to fulfill in commercial warehouses, like, for example, order picking, sorting, packaging, and inspection. However, this work is usually easy learn

$\mathcal{T} = \{1, \ldots, T\}$	Time horizon ($t \in \mathcal{T}$ is one day)
$\mathcal{A} = \{1, \ldots, A\}$	Set of full time workers
$c_a \in \mathbb{R}$	Costs of full time worker per hour
$c_p \in \mathbb{R}$	Costs of part time worker per hour
$c_e \in \mathbb{R}$	Costs of external worker per hour
$h_{\max} \in \mathbb{N}$	Maximum number of working hours for a full time worker per day
$D_{\text{off}} \in \mathbb{N}$	Minimum number of days off of full time workers in a period of length P
$\rho \in \mathbb{R}$	Maximum fraction of part time relative to full time workforce

Table 7.1: Parameter notations

and computer-supported so that all tasks can be performed by all employees which eventually justifies the assumption that there is one department, i.e., one type of work. The real-life case which is used to test the decision support tool confirms this assumption. Lastly, we restrict the staffing problem by the commonly used condition that the ratio part time relative to full time workforce is limited (Bard et al., 2007) which is often used if flexible workers are accompanied with lower costs than full time employees and a certain level of full time staff assignment is desired.

7.2.1 Notations

To formulate the model let $\mathcal{T} = \{1, \ldots, T\}$ denote the discrete time interval which describes the planning horizon. We assume a single shift mode, so that one period $t \in \mathcal{T}$ represents, for example, one day. The warehouse manager can allocate full time employees $\mathcal{A} = \{1, \ldots, A\}$ to specific days $t \in \mathcal{T}$ with the help of the binary variable x_a^t and a specific number of hours by choosing h_a^t for $a \in \mathcal{A}$ and $t \in \mathcal{T}$. As mentioned above full time workers receive a constant salary, which is denoted by the cost parameter c_a stating the costs per worker and hour. The total costs of full time workforce is independent from the shift schedule. This concept describes a base level of workforce that is always available and limited by a maximum number of hours per day and a minimum number of days off in the period. Part time workforce at day $t \in \mathcal{T}$ is denoted by p^t and solely restricted by the ratio $\rho \in [0, 1]$ of workload relative to full time workload. Part time workforce is paid for each hour assigned, their costs are denoted by the cost parameter c_p per hour. Recourse costs are denoted by the cost parameter c_e and the corresponding total number of labor hours is described by the variable y^t. Table 7.1 summarizes the notation of the parameters used. The decision variables are listed in Table 7.2.

$x_a^t \in \{0, 1\}$	Binary variable equals 1 if $a \in \mathcal{A}$ is working at day $t \in \mathcal{T}$, otherwise 0
$h_a^t \in \mathbb{N}$	Number of hours that $a \in \mathcal{A}$ is scheduled for work at day $t \in \mathcal{T}$
$p^t \in \mathbb{R}$	Number of working hours performed by part time workers at day $t \in \mathcal{T}$
$y^t \in \mathbb{R}$	Working hours performed by externals at day $t \in \mathcal{T}$

Table 7.2: Decision variable notations

7.2.2 The deterministic model

We introduce the formal model by developing the deterministic optimization model in which the objective is to minimize the total labor costs along the planning horizon. For the sake of clarification we summarize the constant costs of full time workforce by $C_{\mathcal{A}} = c_a \cdot h_{\max} \cdot T \cdot A$. Further, workforce demand and labor shortage are known parameters in the deterministic model so that we add here the following the notation to Table 7.1:

$d^t \in \mathbb{R}$	Workforce demand at day t
$s^t \in [0, 1]$	Fraction of absent workforce

The corresponding optimization problem which minimizes the total labor cost can be stated as follows:

$$\min TC = \min C_{\mathcal{A}} + \sum_{t=1}^{T} \left(c_p p^t + c_e y^t \right) \tag{7.1}$$

subject to:

$$d^t - \left((1 - s^t) \left(\sum_{a=1}^{A} h_a^t \right) + p^t + y^t \right) \leq 0 \quad \forall t \in \mathcal{T} \tag{7.2}$$

$$h_a^t - (x_a^t h_{\max}) \leq 0 \quad \forall t \in \mathcal{T},\ a \in \mathcal{A} \tag{7.3}$$

$$\sum_{t=1}^{T} x_a^t - T + D_{\text{off}} \leq 0 \quad \forall\, a \in \mathcal{A} \tag{7.4}$$

$$p^t - \rho \sum_{a=1}^{A} h_a^t \leq 0 \quad \forall t \in \mathcal{T} \tag{7.5}$$

The objective function in Equation (7.1) consists of three cost components being the labor cost for full time workers, the labor cost for working hours conducted by part time workers, and the labor costs for external workforce. Constraint (7.2) ensures that the demand is fulfilled at each time step. Constraint (7.3) limits the number of working hours for each full time worker per day. Constraint (7.4) regulates the number of days off within the period $\mathcal{T}$. Constraint (7.5) restricts the amount of work that can be fulfilled by part time workers relative to full time workers at each time step. External workforce cannot be restricted in such a way, since our approach relies on complete recourse modeling, which means that for any realization of the stochastic parameters there is at least one feasible recourse option, which is potentially accompanied with high costs.

This deterministic formulation is not used for experiments in the following, but it assists in developing the multistage stochastic formulation, because it clearly describes the warehouse staffing problem if all information would be available. In the following section we introduce the multistage stochastic models which build the basis for our decision support tool.

7.3 Stochastic optimization models

We use five different multistage stochastic programming approaches that are suited to deal with uncertainties and known from financial risk management and other application areas.

In contrast to the previous deterministic formulation demand and labor shortage are here not known in advance, but described by a bivariate stochastic process $\{d^t, s^t\}_{t \in \mathcal{T}}$ with the realizations $(d_l^t, s_l^t) \in L_t$ for $t \in \mathcal{T}$. $L_t \subset \mathbb{R}^2$ thereby denotes the finite set of realizations at time t. Finiteness of those sets is an important assumption that we have to make, because it enables us to represent the possible outcomes with a scenario tree and to solve the resulting optimization models with traditional solvers.

The first proposed model represents a classical risk-neutral multistage stochastic modeling approach which aims to minimize the expected total costs. We use this model to benchmark the risk averse approaches. Clearly, if any kind of risk aversion is the optimization objective the expected total costs are higher than with this model, but the solutions also provide more protection. Yet too high expected total costs are impractical.

The second and third model are likewise multistage stochastic optimization models, yet here the conditional value at risk (CVaR) and the expected excess, respectively, are minimized. The CVaR and the EE are related and represent two well-suited risk measures for staff planning problems, because both incorporate not only the probability of certain risks but also the extent to which in these cases the expected costs exceed a specific barrier. Both risk measures are often used indicators in financial risk management, because of their mathematical properties. For a more details description of those we refer to, for example Rockafellar and Uryasev (2002), Acerbi and Tasche (2002), an Artzner et al. (1999).

In particular the CVaR and the EE are also suited for mean-risk modeling, which is applied here in the fourth and fifth model. As both risk optimization and expected cost minimization alone might utilize limited information on the stochastic parameters or cost structures, mean risk approaches form an alternative which incorporates both (Heinze, 2008).

To formulate the models we introduce a notation, which describes conditional probabilities of realizations. Let until time t the historical realizations be given by $((d_{l_1}^1, s_{l_1}^1), (d_{l_2}^2, s_{l_2}^2), \dots, (d_{l_{t-1}}^{t-1}, s_{l_{t-1}}^{t-1})) \in L_1 \times L_2 \times \dots \times L_{t-1}$. Then, we denote the conditional probability of the realization $(d_l^t, s_l^t) \in L_t$ to occur at time t by $\pi_{l_t}^{l^{t-1}}$, where $l^{t-1} = (l_1, \dots, l_{t-1})$. In other words $\pi_{l_t}^{l^{t-1}}$ is the probability of a realization (d_l^t, s_l^t) given the information that until time t the realizations $(d_l^1, s_l^1), \dots, (d_l^{t-1}, s_l^{t-1})$ have been observed. π_{l_1} denotes the absolute probabilities of outcomes at the first time steps.

7.3.1 Expected value based multistage stochastic model

The first optimization approach that we incorporate in our analysis is a classical and risk-neutral multistage stochastic programming model. Its objective function is a nested expression of expected costs for each time step, while minimization is targeted over the sum of all realized costs of past time periods and expected costs of future periods. Here, the amount of externals used in time step t are incorporated by the stochastic recourse costs C^t which is a function of stochastic demand and shortage. The first expected-value based model of problem 7.1 can then be stated as

$$\begin{aligned} \min M1 &= \min(C_{\mathcal{A}} + c_p p^1 + \mathbb{E}_1[C^1 + c_p p^2 + \mathbb{E}_2[C^2 + c_p p^3 + \mathbb{E}_3[C^3 + \qquad (7.6) \\ &\quad + \cdots + \mathbb{E}_T[C^T] \dots]]]) \\ &= C_{\mathcal{A}} + c_p p^1 + \sum_{l_1 \in L_1} \pi_{l_1}(C_{l_1}^1 + c_p p^2 + \sum_{l_2 \in L_2} \pi_{l_2}^{l^1}(C_{l_2}^2 + c_p p^3 + \qquad (7.7) \\ &\quad + \sum_{l_3 \in L_3} \pi_{l_3}^{l^2}(C_{l_3}^3 + \cdots + \sum_{l_T \in L_T} \pi_{l_T}^{l^{T-1}} C_{l_T}^T))) \end{aligned}$$

subject to

$$\begin{aligned} C_{l_t}^t &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t \qquad (7.8) \\ C_{l_t}^t - c_e\left(d_{l_t}^t - p^t - (1 - s_{l_t}^t)\sum_{a \in \mathcal{A}} h_a^t\right) &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t \qquad (7.9) \\ h_a^t - (x_a^t h_{\max}) &\leq 0 \quad \forall t \in \mathcal{T},\ a \in \mathcal{A} \qquad (7.10) \\ \sum_{t=1}^{T} x_a^t - T + D_{\text{off}} &\leq 0 \quad \forall\, a \in \mathcal{A} \qquad (7.11) \\ p^t - \rho \sum_{a=1}^{A} h_a^t &\leq 0 \quad \forall t \in \mathcal{T} \qquad (7.12) \end{aligned}$$

Decision variables in this formulation are x_a^t and h_a^t to assign full time workforce to the schedule and p^t for defining the amount of work to be performed by part time staff. Potential shortage in labor causes the recourse costs C^t for $t \in \mathcal{T}$, which are modeled with the help of Constraints (7.8) and (7.9), so that the objective function in model 1 (Equation (7.7)) states the total costs of full time and part time staff together with the expected recourse costs of each time step. The latter formulation is thereby the key of stochastic modeling approaches, since it provides a deterministic ILP formulation, if the number of realizations is finite at each time step. Constraints (7.8) and (7.9) are used to model the recourse costs. In accordance with the deterministic framework Constraints (7.10) - (7.12) guarantee the feasibility of the decisions $(x_1^t, \dots, x_A^t, h_1^t, \dots, h_A^t, p^t)$. In the following we refer to the problem (7.7) as model 1.

7.3.2 Risk based models

When fluctuations in the stochastic outcomes are high, the expected value problem might lead to solutions which are impractical for real applications. If the standard deviation of outcomes is high, any decision policy that solely adjusts to mean values can cause high recourse costs. To protect against the risk of unexpectedly high cost we propose the first two risk averse modeling approaches which are based on the minimization of a the risk measures conditional value at risk (CVaR, model 2) and expected excess (EE, model 3).

For one period models, the conditional value at risk for a level $\alpha \in (0,1]$ expresses the expected cost of unfavorable realizations, i.e, in our case of high recourse costs C^t. Which realizations are thereby unfavorable is defined by a parameter α. We elaborate in Section 7.4.3 on the effects of different values for α. To define the CVaR for one period first let Z be a random variable Z. The conditional value at risk is then defined as follows:

$$CVaR_\alpha = \inf_{z \in \mathbb{R}} \left\{ z + \frac{1}{\alpha} \mathbb{E}\left[(Z-z)^+ \right] \right\},$$

where

$$(Z-z)^+ = \begin{cases} Z - z & \text{, if } Z - z \geq 0 \\ 0 & \text{, otherwise} \end{cases}$$

Put simply, the conditional value at risk for a level α expresses the expected cost, if one of the $\alpha \cdot 100\%$ worst realizations occur. We can transfer the conditional value at risk into a multi-period risk measure of our recourse costs C^t. With $\alpha = (\alpha_1, \dots, \alpha_T) \in (0,1]^T$ the $CVaR_\alpha$-based formulation of our problem becomes (model 2)

$$\begin{aligned} \min M2 &= \min C_{\mathcal{A}} + \sum_{t=1}^{T} z_t + c_p p^1 + \frac{1}{\alpha_1} \mathbb{E}_1 [Z^1 + c_p p^2 + \frac{1}{\alpha_2} \mathbb{E}_2 [Z^2 + c_p p^3 + \qquad (7.13) \\ &\quad + \frac{1}{\alpha_3} \mathbb{E}_3 [Z^3 \cdots + \frac{1}{\alpha_T} \mathbb{E}_T [Z^T] \dots]]] \\ &= \min C_{\mathcal{A}} + \sum_{t=1}^{T} z_t + c_p p^1 + \frac{1}{\alpha_1} \sum_{l_1 \in L_1} \pi_{l_1} (Z_{l_1}^1 + c_p p^2 + \frac{1}{\alpha_2} \sum_{l_2 \in L_2} \pi_{l_2}^{l^1} (Z_{l_2}^2 + \qquad (7.14) \\ &\quad + c_p p^3 + \frac{1}{\alpha_3} \sum_{l_3 \in L_3} \pi_{l_3}^{l^2} (Z_{l_3}^3 + \cdots + \frac{1}{\alpha_T} \sum_{l_T \in L_T} \pi_{l_T}^{l^{T-1}} Z_{l_t}^T) \dots)) \end{aligned}$$

subject to

$$
\begin{aligned}
Z^t_{l_t} &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t & (7.15)\\
Z^t_{l_t} - (C^t_{l_t} - z_t) &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t & (7.16)
\end{aligned}
$$

and subject to the Constraints (7.8), (7.9), and (7.10) - (7.12).

Similar to the above definition the multi-period CVaR is here modeled with the additional variables $z_1, \ldots, z_T$ and the random variables $Z_1, \ldots, Z_T$. Naturally the remaining conditions of the risk model correspond to the expected value based formulation of model 1.

A related measure of risk is the expected excess. Let $\beta \in \mathbb{R}_+$ be a predefined value and B again a random variable with values in $\mathbb{R}$. The expected excess for one period is formally defined by the expected value of the difference between β and all realizations of B greater than β, i.e.

$$EE_\beta = \mathbb{E}\left[(B - \beta)^+\right]$$

Like the CVaR, the EE thus describes the risk of exceeding a specific barrier. In contrast to the CVaR the expected excess defines this barrier with a real number and independent from the number or probability of scenarios for which B exceeds β. Let $\beta = (\beta_1, \ldots, \beta_T) \in \mathbb{R}$ the EE_β-based formulation of our problem can be stated as follows (Model 3):

$$
\begin{aligned}
\min M3 &= \min C_{\mathcal{A}} + c_p p^1 + \mathbb{E}_1[B^1 + c_p p^2 + \mathbb{E}_2[B^2 + c_p p^3 + & (7.17)\\
&\quad + \mathbb{E}_3[B^3 \cdots + \mathbb{E}_T[B^T]]]] \\
&= \min C_{\mathcal{A}} + c_p p^1 + \sum_{l_1 \in L_1} \pi_{l_1}(B^1_{l_1} + c_p p^2 + \sum_{l_2 \in L_2} \pi^{l^1}_{l_2}(B^2_{l_2} + c_p p^3 + & (7.18)\\
&\quad + \sum_{l_3 \in L_3} \pi^{l^2}_{l_3}(B^3_{l_3} + \cdots + \sum_{l_T \in L_T} \pi^{l^{T-1}}_{l_T} B^T_{l_t})))
\end{aligned}
$$

subject to

$$
\begin{aligned}
B^t_{l_t} &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t & (7.19)\\
B^t_{l_t} - (C^t_{l_t} - \beta_t) &\geq 0 \quad \forall t \in \mathcal{T},\ l_t \in L_t & (7.20)
\end{aligned}
$$

and subject to the Constraints (7.8), (7.9), and (7.10) - (7.12).

As indicated above the main difference between the CVaR and the EE to optimize risks, is the definition of risk either by those realizations that belong to a specified percentage of unfavorable realizations (CVaR) or by those realizations that exceed a specific monetary barrier (EE). Both formulations have advantages and disadvantages for specific settings on which we elaborate in Section 7.5. For example, the EE formulation might have no risk aversion impact for high values of β_t, if the potential recourse costs are low. On the other hand, the CVaR formulation might suggest highly expensive over-staffing solutions for low values of α_t, especially if the standard deviation of the recourse costs are high. In this case model 2 would suggest staffing policies that prepare for only a few expensive outcomes which results in high total cost policies.

7.3.3 Mean-risk models

Another option to cope with uncertainties in stochastic planning problems is to combine the expected value problem in model 1 with a risk optimization approach as in model 2 and 3. We propose two options, which rely on the risk measures above. This concept is called mean-risk-modeling (Heinze, 2008), which is the method to model a bi-criteria optimization problem with the objective to minimize a certain risk expression next to the expected value of the objective function.

We begin with the mean-risk model which uses the CVaR (model 4). The CVaR-based mean-risk-model at level $\alpha = (\alpha_1, \dots, \alpha_t)$ is given by

$$\begin{aligned} \min M4 &= \min C_{\mathcal{A}} + \sum_{t=1}^{T} z_t + c_p p^1 + \mathbb{E}_1[C^1 + \frac{Z^1}{\alpha_1} + c_p p^2 + \mathbb{E}_2[C^2 + \frac{Z^2}{\alpha_2} + c_p p^3 + \qquad (7.21) \\ &\quad + \mathbb{E}_3[\cdots + \mathbb{E}_T[C^T + \frac{Z^T}{\alpha_3}] \dots]]] \\ &= \min C_{\mathcal{A}} + \sum_{t=1}^{T} z_t + c_p p^1 + \sum_{l_1 \in L_1} \pi_{l_1} (C^1_{l_1} + \frac{Z^1_{l_1}}{\alpha_1} + c_p p^2 + \sum_{l_2 \in L_2} \pi^{l^1}_{l_2} (C^2_{l_2} + \qquad (7.22) \\ &\quad + \frac{Z^2_{l_2}}{\alpha_2} + c_p p^3 + \sum_{l_3 \in L_3} \pi^{l^2}_{l_3} (C^3_{l_3} + \frac{Z^3_{l_3}}{\alpha_3} + \cdots + \sum_{l_T \in L_T} \pi^{l^{T-1}}_{l_T} (C^T_{l_T} + \frac{Z^T_{l_T}}{\alpha_T}) \dots))) \end{aligned}$$

subject to the Constraints (7.8), (7.9), (7.15), (7.16), and (7.10) - (7.12).

In a similar manner it evolves the EE-based mean-risk model (model 5):

$$\begin{aligned} \min M5 &= \min C_{\mathcal{A}} + c_p p^1 + \mathbb{E}_1[C^1 + B^1 + c_p p^2 + \mathbb{E}_2[C^2 + B^2 + c_p p^3 + \qquad (7.23) \\ &\quad + \mathbb{E}_3[\cdots + \mathbb{E}_T[C^T + B^T] \dots]]] \\ &= \min C_{\mathcal{A}} + c_p p^1 + \sum_{l_1 \in L_1} \pi_{l_1} (C^1_{l_1} + B^1_{l_1} + c_p p^2 + \sum_{l_2 \in L_2} \pi^{l^1}_{l_2} (C^2_{l_2} + B^2_{l_2} + \qquad (7.24) \\ &\quad + c_p p^3 + \sum_{l_3 \in L_3} \pi^{l^2}_{l_3} (C^3_{l_3} + B^3_{l_3} + \cdots + \sum_{l_T \in L_T} \pi^{l^{T-1}}_{l_T} (C^T_{l_T} + B^T_{l_T}) \dots))) \end{aligned}$$

subject to the Constraints (7.8), (7.9), (7.19), (7.20), and (7.10) - (7.12).

We discuss in the following sections the impact of each modeling approach on the resulting staffing policies in specific settings. Mean-risk approaches might be more applicable to situations in which either the expected recourse costs are so high that not much attention can be given to risk protection, so that the policy design has to concentrate more on expected outcomes, or there is not much uncertainty so that a slight risk aversion together with the minimization of expected cost yields the best results.

7.4 Experimental design

In order to analyze the behavior of the different models and to shape the decision support tool for staff scheduling policies we solve all five optimization models for various demand and shortage situations. Since the corresponding deterministic programming model of a multistage stochastic model can become extremely large (with increasing time horizon, number of scenarios, or number of constraints of the underlying problem), we used an approximation procedure to create scenario trees, developed by Heitsch and Römisch (2009), which can reduce the dimension of potentially large scale input data in order to make the decision support tool applicable for a wide range of practical staffing problems. In Section 7.4.1 we explain this procedure in more detail. Section 7.4.2 is used to present four different scenario sets, each of which with specific characteristics that help to examine the behavior of the optimization models. In Section 7.4.3 we calibrate the risk measures to find suitable values for the scalars α_t and β_t, $t \in \mathcal{T}$ and we illustrate how different values for these control parameters affect the risk model solutions.

For all experiments with generated input data we used the following parameters. We consider a setting with 20 full time workers, who can work 5 out of $T = 7$ days (i.e. $D_{\text{off}} = 2$), with $h_{\max} = 8$ hours at maximum per day. Similar to the warehouse operation problems discussed earlier in this thesis, we set the costs of a full time worker to 32€ per hour (similar to Chapter 5). Part time workforce can be employed with up to $\rho = 50\%$ of scheduled full time workforce at each day. The costs of part time labor are 35€ per hour. Under-staffing by full and part time labor has to be recovered with external workforce (overtime, delay costs or the like), which is accompanied with costs of 50€ per labor hour. The number of full time workers and the value of the control parameter ρ are determined by the specific warehouse situation. Here, those values have been selected to match the demand scenarios which we present in Section 7.4.2. Last, a scaling factor is used in the tree approximation to create similar units in the bivariate stochastic process $\{d^t, s^t\}_{t \in \mathcal{T}}$. It is defined by the average demand divided by the average shortage and is used only during the scenario tree construction to facilitate that both demand and shortage are equally affected by the approximation.

7.4.1 Scenario tree construction

For a detailed description of the backward tree construction method we refer to Heitsch and Römisch (2009). The authors suggest a tool to approximate multivariate input data, represented as data fans, by scenario trees with a reduced number of nodes. Thereby data observations are aggregated, if their gap (for a suitable norm) falls below a certain tolerance. The approximation is conducted backwards in time. In our case the procedure is controlled by two parameters $\varepsilon_{T+1} \in \mathbb{R}$ and $q \in [0,1]$. q is used to inductively define those maximum gaps for partial time horizons, whereas ε_{T+1} thereby defines the maximum gap between the original data set and the first approximation step. Formally, the ε_t for $t = 1, \ldots, T$ are defined by

$$\varepsilon_t = q \cdot \varepsilon_{t+1}.$$

Aggregation of two observations takes place by deleting one observation, while its probability is added to the residual observation. Let $l_1^{del}, \ldots, l_K^{del}$ denote the observations which are deleted in step t, then this reduction fulfills the condition

$$\sum_{\tau=1}^{t} \sum_{k=1}^{K} \pi_k \cdot \min_{l_\tau \in L_\tau} \left\{ \sqrt{|d^\tau_{l_k^{del}} - d^\tau_{l^\tau}| + |s^\tau_{l_k^{del}} - s^\tau_{l^\tau}|} \right\} \leq \varepsilon_t$$

is fulfilled. π_k thereby denotes the absolute probability (not conditional probability) of the observation and the minimum expression describes the smallest gap that a deletion of this observation can result in. The outcome of this approximation is a scenario tree with a reduced number of paths, which branches at nodes where the backward tree construction deleted observations.

For our experiments we performed initial tests with original input data consisting of 200 observations. The average demand is set to 170 hours per day with a standard deviation of 30 hours and an average labor shortage of 5% with a standard deviation of 2%. Approximations were generated for all possible combinations of the values q and ε_{T+1} in the ranges $q = 0.975, 0.95, 0.925, \ldots, 0.7$ and $\varepsilon_{T+1} = 1, 2, \ldots, 15$. We solved model 1 for the original data with 200 observations and for each approximation and we are interested in the gaps between optimal solutions of approximated data and original data. From all solutions we distracted the constant costs of full time workforce to analyze the percentaged gaps of the variable costs only.

These preliminary experiments show very small gaps for nearly all values of q and for start values $\varepsilon_{T+1} \leq 10$ (less than 1%). The lower the start value ε_{T+1} the fewer observations are aggregated and the smaller the gap of the resulting scenario tree to the original data. On the other hand small values of ε_{T+1} also cause high dimensional problems which almost reproduce the original input data and might still be difficult to solve. Secondly, we observe that larger values of q produce higher gaps, because if q is set very high, large gaps are tolerated for very short data paths which in turn leads to aggregations of observations that differ significantly. For the remaining experiments we proceed with $\varepsilon_{T+1} = 8$ and $q = 0.8$. This approximation shows a significant reduction in the number of nodes, while a good reflection the original input data is maintained. All models were solved with CPLEX in less than one minute.

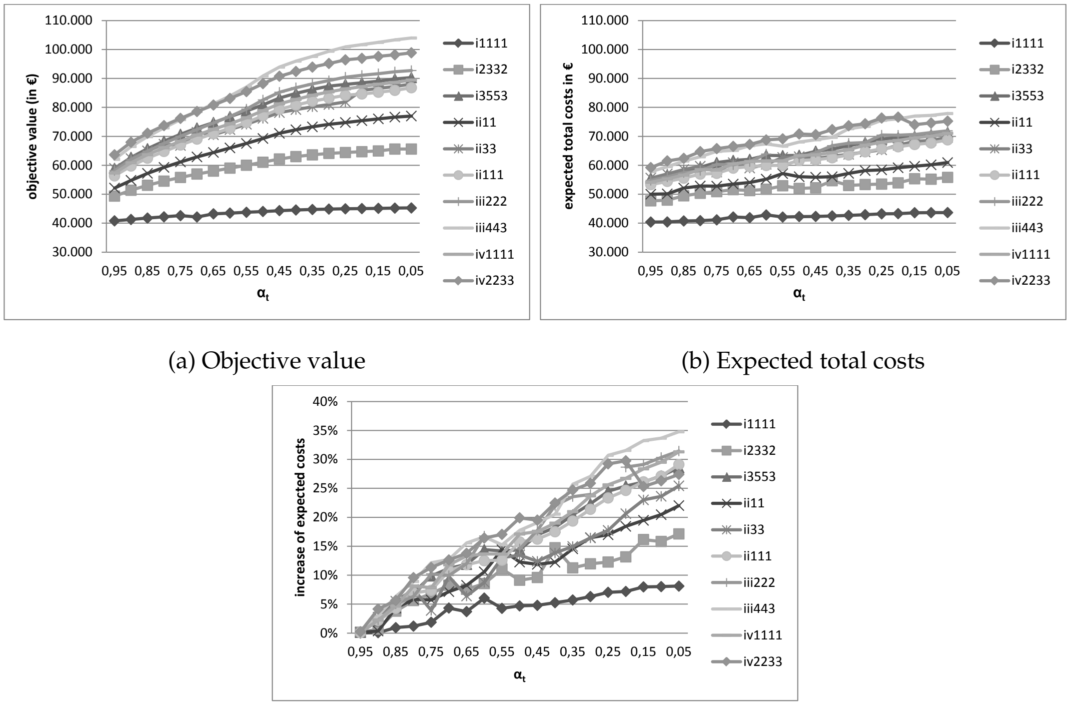
7.4.2 Numerical data scenarios

An overview of the scenario sets that we studied is given in Table 7.3. For simplification we introduce a notation based on the positions in Table 7.3 to refer to a single scenario. For example, 'i1231' refers to the scenario in set i with an average demand of 130 hours,with a standard deviation of 10 hours, an average shortage of 0.02 with a standard deviation of 0.01.

Overall, our aim is to study the behavior of the risk models for stochastic workforce demand and shortage. Thus, with our first set of scenarios (set i) we analyze how different characteristics, such as average and predictability of demand and shortage, affect the solutions of the different models. The total number of different scenarios in set i is 225. The second set, set ii, describes the situation that certain workforce peaks can be predicted. In some practical situations peaks in demand are commonly expected (e.g., specific week day). We differentiate in data set ii the intensity by which the peaks increase the workforce demand on average, i.e. a peak intensity of 0.15 with a regular demand of d_{reg} will result in an average workforce demand of $1.15 \cdot d_{reg}$ on peak days. The standard deviation of the peak intensity in constantly set to 0.05, which describes a relatively predictable situation in which solely fluctuation in workforce demand is high. The total number of scenarios in set ii is 9. Set iii deals with the situation of promotion offers which can significantly increase the workforce demand for a certain period, while promotion start and intensity of the additional workforce demand are stochastic. On average the promotion starts at day 4, while we vary the standard deviation of the start day. Special offers, the adoption of discounts and vouchers by customers, and actions of competitors make promotion offers being accompanied with more uncertainty in workforce demand than on a peak day. Set iii therefore varies not only in the mean of the intensity of the promotion, but also in the corresponding standard deviation, which results in a total number of 48 scenarios. Lastly, set iv combines peaks and promotion days to a certain extent which adds 36 scenarios. In total we study the behavior of the risk models for 318 scenarios, which cover a wide range of warehouses situations and thereby represent a suitable sample to derive the decision support tool rules which assist in warehouse staffing decisions.

Set	Fixed	Position 1	Position	Position 3	Position 4
i		demand mean	demand std	shortage mean	shortage std
		130 (1)	5 (1)	0 (1)	0.01 (1)
		150 (2)	10 (2)	0.01 (2)	0.02 (2)
		170 (3)	20 (3)	0.02 (3)	0.03 (3)
			30 (4)	0.03 (4)	
			40 (5)	0.04 (5)	
ii		# peaks	peak intens mean		
	demand mean 150	1 (1)	0.15 (1)		
	demand std 30	2 (2)	0.25 (2)		
	shortage mean 0.03	3 (3)	0.35 (3)		
	shortage std 0.03				
	peak intens std 0.05				
iii		tstart std	prom intens mean	prom intens std	
	demand mean 150	1.0 (1)	0.15 (1)	0.05 (1)	
	demand std 30	1.5 (2)	0.20 (2)	0.10 (2)	
	shortage mean 0.03	2.0 (3)	0.25 (3)	0.15 (3)	
	shortage std 0.03	2.5 (4)	0.30 (4)		
	tstart mean 4				
iv		# peaks	peak intens mean	tstart std	prom intens mean
	demand mean 150	1 (1)	0.15 (1)	1.0 (1)	0.15 (1)
	demand std 30	2 (2)	0.25 (2)	1.5 (2)	0.20 (2)
	shortage mean 0.03			2.0 (3)	0.25 (3)
	shortage std 0.03				
	tstart mean 4				
	peak int.std 0.05				
	prom intens std 0.05				

Table 7.3: Scenario sets



(a) Objective value

(b) Expected total costs

(c) Increase of expected total costs

Figure 7.2: CVaR minimization for various scenarios and α-levels

7.4.3 Risk measure scaling

The risk measures CVaR and EE are both risk measures that provide insights into the expected costs of negative outcomes. To clarify how different values of $\alpha_1, \dots, \alpha_T$ and $\beta_1, \dots, \beta_T$ affect the staffing policies we selected a few scenarios of each set. Those scenarios we solved for model 1, as well as for the models 2 and 3 for various values of α_t and β_t for $t = 1, \dots, T$. In multi-period models obviously differing values for different time steps are possible for these control parameters. However, such detailed experiments to prioritize single time steps in the planning horizon are an issue of specific case applications and would not add insights for the design of the decision support tool. Hence, we set in all experiments all values α_t and β_t, respectively, to equal values for all $t = 1, \dots, T$. We denote them by $\alpha = \alpha_1 = \dots = \alpha_T$ and $\beta = \beta_1 = \dots = \beta_T$. To illustrate the behavior of the CVaR we depict the results for the solution of model 2 in Figure 7.2.

Model 2 minimizes the expected costs for cases in which workforce demand and shortage realize in the $\alpha \cdot 100\%$ highest recourse costs. Thus, it is important to mention that Figure 7.2a does not depict the actual expected staffing costs, but the expected costs of those most expensive realizations only. These costs are aimed to be minimized by model 2. Obviously, $\alpha =$

1 results in an optimization for the worst 100% of all realization, and thus, in an optimization of the expected total costs as in model 1. For a comparison we show in Figure 7.2b in which expected total costs the corresponding staffing policy would result in.

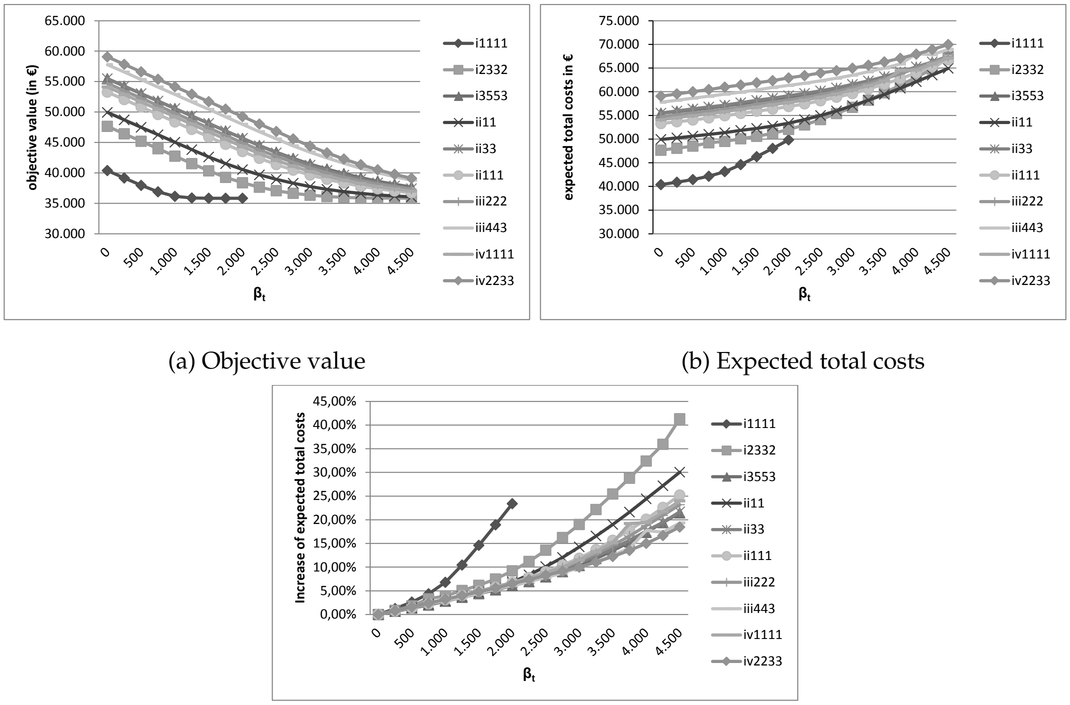
The results show that for decreasing α both the objective value and the expected total costs increase. The objective value increases, because the expected costs of a decreasing number of expensive realizations is considered. The expected total costs increase, since the staffing policy becomes more adjusted to these outcomes, which results in over-staffing policies.

The results already indicate that model 2 is suited for situations with lower potential recourse costs. Especially for such scenarios, as i1111, i2332, and ii11, the increase of costs appears to be lower than for scenarios with higher expected recourse costs, as the scenarios i3553, iii443, and iv2233. More precise, Figure 7.2c depicts the impact of risk protection on the expected total costs. While an $\alpha = 0.05$, i.e. a staffing policy suited for the 5% most expensive realizations can be achieved with $8,11\%$ increased expected total costs for scenario i1111, an increase of $28,56\%$ has to be expected for scenario i3553 where workforce demand and shortage are high with high standard deviations. In general, it can be noted that especially shortages in workforce have a large impact on costs. For high recourse cost situations, such as iii443 and iv2233, risk protection for $\alpha = 0.8$ already approaches 10% of cost increase, so that an optimization of even lower risk levels becomes impractical.

In Figure 7.3 we demonstrate the impact of using model 3, i.e., when the expected excess is the underlying risk measure. The EE denotes the expected costs above the barrier β of all scenarios in which they exceed β. In contrast to the conditional value at risk, model 3 thereby defines risk by explicit costs. Figure 7.3a depicts the objective value of model 3. It decreases with increasing β since the objective function only measures the recourse costs that arise over and above β_t for each day. We thus have with higher β less additional costs even though the overall expected costs (depicted in Figure 7.3b) increase.

For scenario i1111 this model cannot provide more risk aversion than with $\beta = 1500$ where the objective value reaches 35840€ which are the constant costs of full time employees. The corresponding expected total costs of this staffing solution are $46288,37$€ which is 15% above the expected total costs of the risk-neutral approach (model 1). In contrast to the CVaR model, model 3 shows higher increases for low potential recourse cost situations (e.g., scenario i1111) than for potentially higher recourse costs as iv2233.

Figure 7.3c shows the increase of expected total costs in more detail. With increasing β particularly the scenarios i1111, i2332, and ii11, which showed low and mediate expected cost increases for model 2, here show enormous cost increases. The reason for this is that only a few observations in these scenarios show potentially high recourse costs (which exceed β)



(a) Objective value

(b) Expected total costs

(c) Increase of expected total costs

Figure 7.3: Expected excess minimization for various scenarios and β-levels

and to those model 3 adjusts the staffing policy, which leads to unnecessary over-staffing. In contrast, in higher expected recourse cost scenarios also expensive outcomes are more likely, which are continuously respected in the optimization also for lower β which overall results in this a slower increase of expected total costs.

For the remaining experiments to solve all 318 scenarios we proceed with the values of $\alpha = 0.9$ and $\beta = 500$, since these values provided reasonable results for all scenarios. More risk aversion would result in expensive over-staffing policies. In contrast, lower risk aversion produce a nearly risk neutral optimization for many of our scenarios, so that we limit the scope of our analysis to this parameter selection.

7.5 Computational Findings

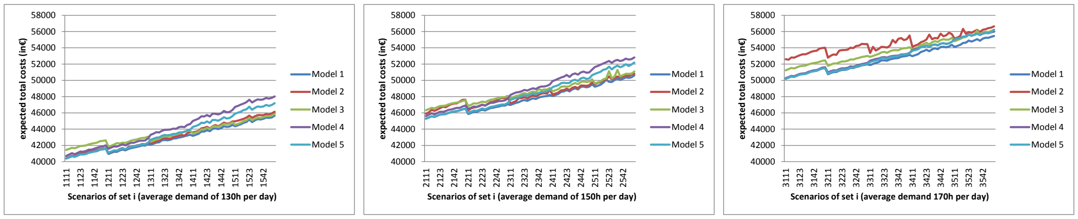
In this section we first discuss the results of solving the four data sets with the five optimization approaches with the aim to identify the implications for a risk control tool with respect to specific warehouse situations. Eventually, Section 7.5.2 is dedicated to present the step-by-step tool to decide on a suitable model to control for risks in warehouse staff scheduling.

Obviously, when comparing the total expected costs for the various models, model 1 will always provide the lowest cost solution, as those are directly minimized in model 1. As already seen in the previous experiments the risk optimization approaches sometimes differ to a large extent from this risk-neutral solution that model 1 determines. Such high differences, however, imply high over-staffing solutions, which would be impractical to implement. On the other hand we will also discover some scenarios in which one or the other model has no impact on the policy compared to model 1, i.e. no risk-aversion effect, and is therefore unsuited for these scenarios as well. The aim of the following analysis is therefore to identify those situations in which the risk and mean-risk approaches provide similar (i.e., slightly higher) expected total costs than model 1 in order to derive a practical risk aversion strategy.

7.5.1 Analysis of models

The results of scenario set i are illustrated in Figure 7.4, separated into the scenarios with small, medium, and high average workforce demand (which also reflect the extent of the potential recourse costs). A more detailed illustration is provided in Appendix A.

The models show very different behavior for the various scenarios in set i. Generally, we can derive in line with previous literature (Zhang et al., 2009) that high average labor shortage as well as uncertainties related to labor shortage significantly increase the expected costs for



(a) average demand of 130h

(b) average demand of 150h

(c) average demand of 170h

Figure 7.4: Results of scenario set i

all five models. This effect can also be observed for scenarios with the same value for the average of labor shortage and solely increasing standard deviation. The same applies when the uncertainty on workforce demand increases.

Model 2 provides reasonable expected total costs when the average workforce demand and thereby the potential recourse costs are low. The difference between the objectives in model 2 and 1 is smaller for scenarios with low recourse costs than for situations with higher workforce demand and therewith higher recourse costs. Furthermore, we can observe with respect to CVaR-based risk measurement that for low recourse costs scenarios model 2 provides lower costs than its mean-risk equivalent model 4. The mean-risk model 4, in turn, appears to be suited for situations with higher average workforce demand and for lower standard deviations, while the effect of risk optimization disappears for model 4 when the standard deviation is too low (e.g. for scenario i3111 with a cost difference of 0.18% compared to model 1). In summary, we find that model 2 is suited for situations in which low recourse costs are expected (e.g., low demand relative to the available workforce, low labor shortage). Its mean-risk variant model 4 instead is a generally save option for low uncertainty settings, while it might have no effect when the potential recourse costs are high and variation is too low, since in these cases the risk measurement part in the objective has only little impact compared to the risk-neutral component in the objective (see Equation (7.21) on page 109).

The EE-based models 3 and 5 show nearly an opposite behavior. First, it appears that model 3 is not a good choice for low potential recourse costs and low variation scenarios, where it produces highly expensive solutions, although these are intuitively the ”easiest” scenarios, because they are almost deterministic. In those situations model 3 creates policies which prepare for cases where the daily recourse costs exceed 500€ (i.e. the costs for external workforce) and those hardy occur. Model 3 becomes more applicable when demand uncertainty increases, since then also the number of scenarios with higher recourse costs increases. In total, model 3 is suited for situations with higher recourse costs, because the risk level is defined by a mone-

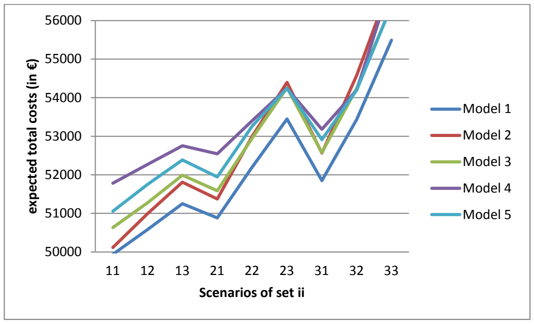


Figure 7.5: Results of scenario set ii

tary value rather than by scenario probabilities. Model 5, in contrast, provides generally good results for small and medium variation, while the effect of risk-protection might disappear for settings with too low uncertainties. In these cases the excess barrier β is hardly reached so that the results of model 5 almost coincide with those of model 1.

Lastly, it appears that for high potential recourse costs and low uncertainty (i.e., standard deviation) both mean-risk models have lower costs than their risk-based alternatives. The risk aversion has little impact in these scenarios, so that the expected costs are similar to those that model 1 yields. On the other hand it stands out that when recourse costs decrease and uncertainty increases, the mean-risk models suggest more expensive staffing policies than their risk-based equivalents, which is somewhat surprising, as the expected total costs are also part of the minimization objective in model 4 and 5. The reason for that is that for these cases too much attention is given to minimizing the risk term in the objectives (7.21) and (7.23), so that the objective value reaches its minimum for higher expected total costs than a sole minimization of the expected total cost would do. Mean-risk models can therefore only be applied when the right trade-off between the terms in the objective is given. They are not suited for situations with too low uncertainties, because then the risk term has no impact and the solution is almost risk neutral, and for situations with too high uncertainties, because then the risk term has too much impact and the solution becomes expensive. For such situations a scaling factor between the two terms in the objective could help to make a mean-risk model applicable. However, in this work we do not consider such scaling factors since they would shift the focus of the objective of the mean-risk model either toward the risk or toward the risk-neutral objective which our decision support tool would correspondingly suggest.

The results of scenario set ii are summarized in Figure 7.5. We find that the risk based models contentiously provide good results throughout all scenarios (1.42% higher costs with model 2 and 1.45% higher costs with model 3 on average). Model 2 becomes slightly more expen-

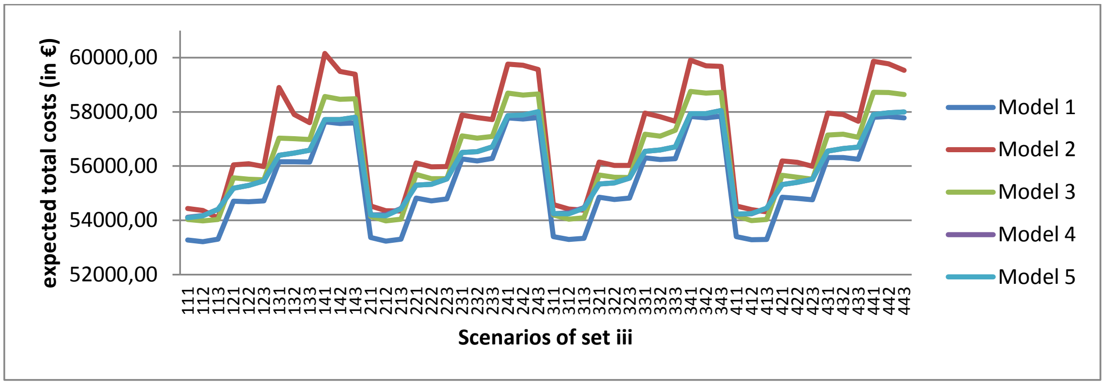


Figure 7.6: Results of scenario set iii

sive (2.75% higher costs for scenario ii33), when the average peaks intensity increases, which again represents a scenario with higher recourse costs. This effect can be explained by similar reasons as in the analysis of scenario set i. The CVaR optimization affects a large percentage of expensive scenarios when uncertainty increases. Although this also implies the more risk averse solution strategy, model 2 is in these cases a cost-intensive option. The basic setting of set ii, described by the fixed parameters in Table 7.3 on page 113, characterizes a situation with relatively high fluctuations in workforce demand and shortage throughout all scenarios of set ii, which explains why the mean-risk models provide expensive staffing solutions for the scenarios ii11-ii22. For the remaining scenarios model 4 and 5 almost coincide with the risk models.

The solutions of scenario set iii, depicted in Figure 7.6, suggest the following: The mean-risk models 4 and 5 almost coincide throughout all scenarios. Obviously, with an increase of the average intensity of a promotion event the expected total costs increase for all models. No significant impact can be observed for an increase of the standard deviation of the promotion intensity, which might be explained by the selection of rather small values of this deviation in set iii. The uncertainty of the promotion start day has an effect on the expected total costs and also on the risk model outcomes. The effect, however, is very small relative to the impact that the intensity of promotions has on outcomes. The reason for this might be that in such high recourse cost settings no full time staff can be additionally assigned to prepare for an earlier start of a promotion. Then, this can only be arranged by additional part time staff at this one day, which causes lower costs than in cases in which the overall costs are also bounded. In total set iii again describes a relatively stable situation with little uncertainty. In line with the analysis of set i we thus observe that the mean-risk models provide solutions with lower costs than the risk models, and, moreover, that for this setting the EE-based model 3 provides

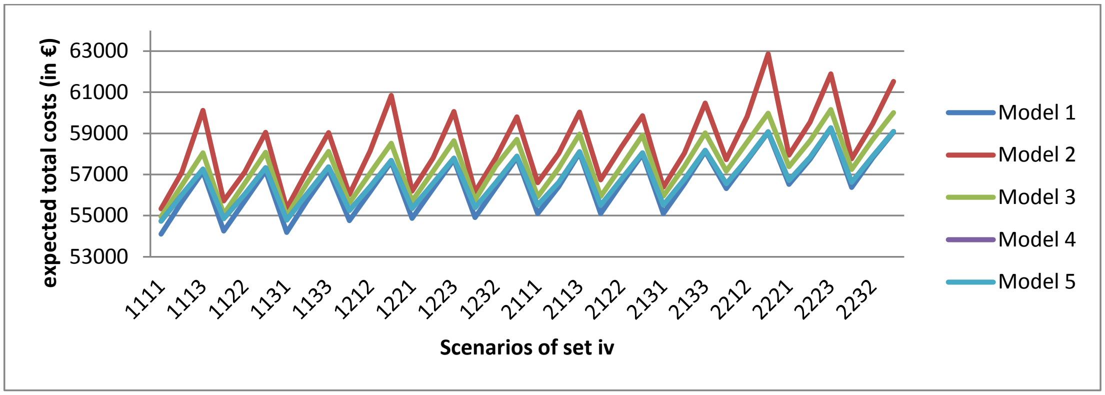


Figure 7.7: Results of scenario set iv

solutions at lower costs than the CVaR-based model 2, which confirms the results of set i in which the EE-based approach outperforms the CVaR approach for high recourse cost settings.

Lastly, we analyze the behavior of the risk optimization approaches for scenario set iv, for which we depict the results in Figure 7.7. Clearly, there is an increase in expected costs, when the number of peak days increases and also with an increase of the workload due to promotions. Again, we observe that the impact of uncertainty related to the exact start of promotion offers is present, but not significant. The results of the mean-risk models 4 and 5 again do not differ significantly. We also find, similar to the previous case, that the risk models have higher expected total costs than their mean-risk alternatives throughout all scenarios, as the fluctuation of workforce demand is low. The same applies to the intensity of additional workload through peaks. Overall, scenario set iv represents an workforce intensive case, for which the expected excess models appear to be the best option.

These analyses showed that the choice of a risk control approach in warehouse staff situations is mainly dependent on the level of uncertainty and the extent of the potential recourse costs of the staffing situation. Special uncertainty sources, such as peak days and labor effort through promotion offers, fit in this concept and influence the outcome of risk optimization mainly based on their impact on costs and their predictability. We therefore propose a risk approach determination based on these two factors and therewith suggest the selection of a risk optimization approach in accordance with the decision matrix depicted in Figure 7.8.

We do not specify precise barriers for the quarters in the matrix, because these can result from more than one source in practical applications. Also in our considered warehouse situations uncertainty was driven by labor demand and shortage. In our experiments low uncertainty reflected very stable situations with a labor fluctuation of 5 hours and a shortage fluctuation of 1%. The highest uncertainty was given by scenarios with 40 hours labor demand

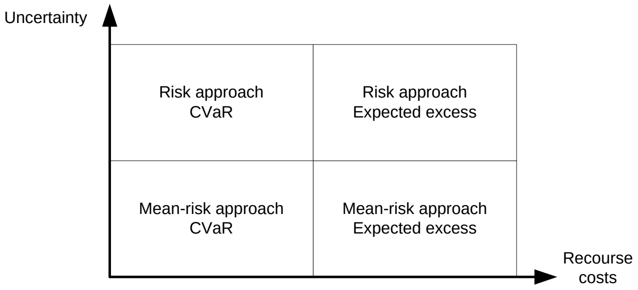


Figure 7.8: Decision matrix for risk optimization approach

fluctuation and a shortage standard deviation of 3%. Moreover, the selection of an approach is not exclusive; applications with medium uncertainty and recourse cost situations might allow for two risk optimization approaches with a adjacent models in the matrix.

7.5.2 Decision support tool for risk management

We design a decision support tool to identify an appropriate risk measurement approach for a specific warehouse case. The tool also includes the specification of the warehouse situation in order to clarify the underlying optimization model. Questions on uncertainties and their impact thereafter guide the decision on which model should be used and how to calibrate it for the specific purpose of the warehouse manager.

Step 1 *Identification of the underlying optimization problem*

Specify the desired planning horizon T. Specify potential fixed labor costs in the planning horizon (costs which arise independent from specific workforce schedules), set variable costs (i.e., labor costs per hour, which are only paid when workload is assigned), and other values and parameters related to the problem. Which constraints shape scheduling problem as exemplary shown in Section 7.2.2 (e.g., multiple shifts, shift lengths, and breaks)?

Step 2 *Determine recourse options*

Which options are available to recover shortage in labor on short notice? Examples are overtime, employing external workforce, postponement or the like. Multiple options are possible; yet each shortage in workforce demand must be recovered by one or another option. Specify the costs of each recourse option and potentially limitations which have to be added with the help of constraints (e.g., overtime limits).

Step 3 *Compose the resulting optimization model*

Combine the scheduling problem of Step 1 with the recourse options of Step 2 into one single optimization model as exemplary shown in model 1 in Section 7.3.1.

Step 4 *Analyze historical data on workforce demand and potentially labor shortage*

If historical data, especially for workforce demand, shows no significant trend, which suggests that future observations considerably differ from past observations, a sufficiently large set of historical data can be used to determine labor schedules for the future.

If historical data shows respective trends a suitable forecasting method which incorporates those trends has to be used to simulate future demands.

Step 5 *Develop scenario trees*

Depending on the number of data observations, the complexity of the model (i.e., number of constraints), and the length of the planning horizon T, it might be necessary to reduce the number of observations with an approximation method as suggested in 7.4.1.

Step 6 *Analyze data*

Determine means and standard deviations of the uncertainty sources, for example, workforce demand, shortage, peak days intensity, promotion effort intensity.

Step 7 *Decide on a risk optimization approach*

Determine the most suitable approach by locating the scenario in the matrix in Figure 7.8. If no unique quarter can be chosen, proceed with the most suitable options in Step 8 and make a final decision based on experimental tests.

Step 8 *Calibrate control parameters*

If in Step 7 a CVaR-based model has been selected, an $\alpha_t \in (0, 1)$ has to be defined for each time step $t = 1, \ldots, T$. High values of α_t create lower risk aversion, and thereby also lower total costs, than low values of α_t. Recall that $\alpha_t = 0.9$ realizes an optimization of the 90% highest recourse cost observations in the data set.

If in Step 7 an EE-based model has been selected, choose a β_t for each $t = 1, \ldots, T$. β_t has to be a monetary value and determines the excess barrier. Recall that $\beta_t = 500€$ leads to an optimization which minimizes the average occurring costs on days were 500€ recourse costs are exceeded.

Step 9 *Test the selection experimentally*

Solve the model(s) selected in Step 7 with the control parameters selected in Step 8. Compare the results with the expected total costs and analyze the resulting policy.

If the results do not match the desired risk aversion level go back to Step 8 and increase the risk aversion (lower α_t and higher β_t) until an acceptable solution is found.

If the results exceed the possible total costs go back to Step 8 and decrease the risk aversion (higher α_t and lower β_t) until an acceptable solution is found.

7.6 Case application

To apply the decision support tool we consider the staffing problem of a Dutch commercial warehouse and study the behavior of the risk aversion approaches for this real-life case. Doing so, we can clarify how to determine the warehouse situation according to the decision matrix in Figure 7.8 and test the corresponding optimization approach in contrast with the others.

7.6.1 Warehouse setting

The warehouse handles order requests placed by individual consumers as well as business to business and has therefore manual as well as semi-automated order picking implemented. The warehouse has full time and part time staff members employed, who work eight and four hours per day, respectively, when assigned to a shift. Potential labor shortage is recovered with external workforce via a temporary employment agency; postponement of customer requests is not possible and delays are aimed to be avoided. Work is conducted in two shifts and five days a week. However, the specific staff assignment in early and late shift in this warehouse is separated from the decision on the number of labor hours to allocate for each day. Therefore we also focus on the number of full time and part time hours to be allocated only.

The planning horizon of the company is one week. The costs of full and part time employees are 25€ per hour. External workforce is paid with 19€ per hour. Yet external workers are usually not familiar with the warehouse and the work, so that those are considered to be less productive than the personnel employed by the warehouse. To account for this imbalance we assume a productivity rate of 0.7 for external workforce. Similar to the experiments in the previous sections, also in this real-life case different types of work are not distinguished, as the warehouse has a cross-train policy implemented, which implies that each full time and part time employee is capable of performing each type of work.

Workforce demand is derived from incoming transactions of one year. The labor effort differs depending on whether an order consists of a single order line, a few order lines, or whether a request is fulfilled for a business partner and bulks have to be processed. We estimated the resulting demand of labor hours with a similar procedure as the company on the basis of the

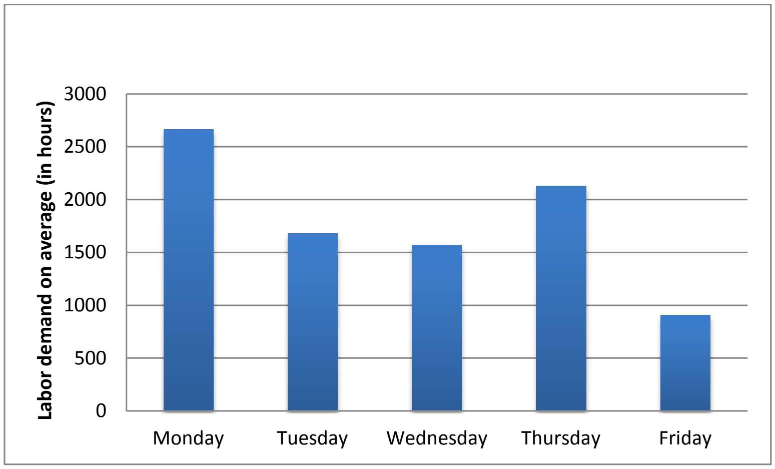


Figure 7.9: Average labor demand for single week days

number of transaction per day. The average labor demand per day is 1785 hours with a high standard deviation of 41%. Yet this average covers the entire year. The work load for specific weekdays, however, varies significantly, so that the demand (except for this predictable variation for the week days) shows a less uncertain behavior if we consider it for each day of the week. The average workforce demand throughout the week is depicted in Figure 7.9. The standard deviation for the single weeks days then constitutes 24% on average. No exact historical data on absenteeism of workers is available; however, it has been observed to be approximately 2% on average with a low standard deviation (1%).

7.6.2 Decision tool

Step 1 *Identification of the underlying optimization problem*

The planning period of the company is one week and the warehouse is operating 5 days per week, which results in $T = 5$ time steps. Staff is paid on an hourly basis. The staff members consist of full time employees, who work eight hours when being assigned to a shift and part time employees who work four hours per shift. The costs of a labor hour are $c_w = 25$€ for full and part time staff members.

Step 2 *Determine recourse options*

Postponement or delays are not possible, so the only allowed recourse option is the recovery of labor shortage with external workforce, with a productivity of 0.7 relative to full time and part time staff and with costs of $c_e = 19$€ per labor hour.

Step 3 *Compose the resulting optimization model*

We denote the number of full time workers assigned at day t by x^t_{ft} and the number of part time workers by x^t_{pt}. The risk-neutral model of this case has the following form:

$$\min c_w(x^1_{ft} + x^1_{pt}) + \mathbb{E}_1[C^1 + c_w(x^2_{ft} + x^2_{pt}) + \mathbb{E}_2[C^2 + \cdots + \mathbb{E}_T[C^T]\ldots]]$$

subject to

$$\begin{array}{rcll} C^t_{l_t} & \geq & 0 & \forall t \in \mathcal{T},\ l_t \in L_t \\ C^t_{l_t} & \geq & \frac{c_e}{0.7}\left(d^t_{l_t} - p^t - (1 - s^t_{l_t})\left(8x^t_{ft} + 4x^t_{pt}\right)\right) & \forall t \in \mathcal{T},\ l_t \in L_t \end{array}$$

Step 4 *Analyze historical data on workforce demand and potentially labor shortage*

In the current practice of this warehouse historical data is used to predict future demand. Historical transaction data of one year is used, while the number of transactions is translated in labor hours similarly to the policy of the warehouse. Labor shortage is simulated according to the past observations with an average of 2% and a standard deviation of 1%.

Step 5 *Develop scenario trees*

The data did not require an approximation and could be directly used in the optimization, since the time horizon was relatively short and the available data fan could be used.

Step 6 *Analyze uncertainty sources*

The warehouse situation described here shows a scenario with a relatively highly fluctuating workforce demand in comparison with the previous experiments (approximately 24%). The potential recourse costs can be considered to be low, since the cost difference between pre-scheduling costs (25€ per labor hour) and recourse costs ($\frac{19}{0.7}$ € per labor hour) is rather low. In contrast to the warehouse example from the numerical experiments, in which part time labor use (pre-scheduling costs) was also bounded and thereby high recourse costs occurred, here the warehouse could prepare for any outcome before the realization, which naturally lowers the recourse costs.

Step 7 *Decide on a risk optimization approach*

Given the analysis of the case as described above our decision support tool suggests to use a risk approach rather than a mean-risk model, and, given the low recourse costs, a CVaR approach rather than an EE-based model (see Figure 7.10).

Step 8 and 9 *Calibrate control parameters and test the selection experimentally*

We summarize the last two steps by providing an overview of the outcomes that various values for α in the CVaR-based risk modeling approach as well as of the outcomes of various values for β in an EE-based risk approach lead to. The total expected cost of

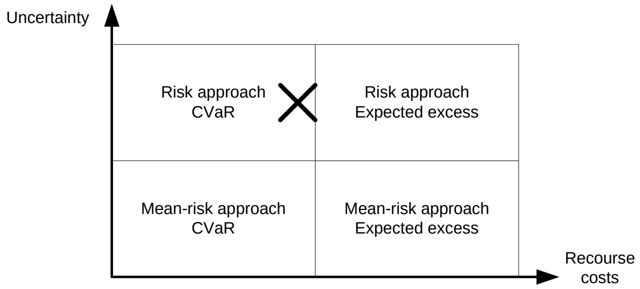
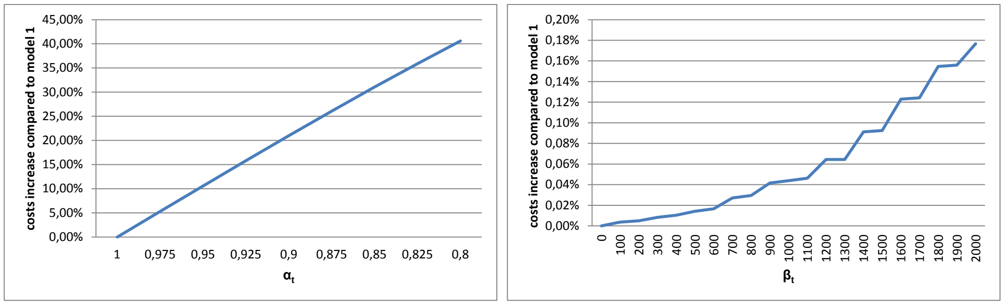


Figure 7.10: Risk approach selection for case application

the risk neutral approach (i.e., model 1) are 48967.47€. We include model 3 also in this analysis since it could potentially also be an option for this warehouse case and illustrate the expected total cost of model 2 and 3 by demonstrating the cost increase in comparison with model 1. Since a risk averse decision making can hereby be quantified, the final selection of the risk-aversion level (i.e. α and β, respectively) can now be made by the decision maker. Figure 7.11 shows the results of both risk models for a number of values of α and β, respectively.



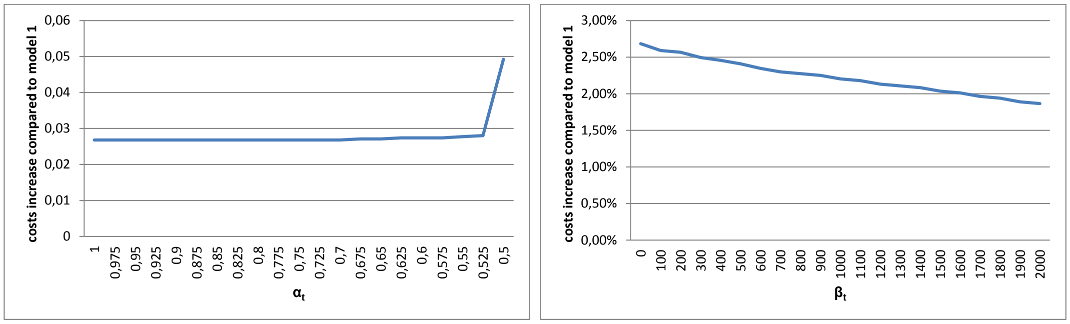
(a) CvaR optimization for various values of α

(b) EE optimization for various values of β

Figure 7.11: Cost increases with risk-based modeling relative to risk-neutral optimization

7.6.3 Case results and implications

The decision on the warehouse situation which the real-life case reflects was made by analyzing the uncertainty sources and recourse costs in comparison with the numerical examples that we studied in the previous section to derive the decision tool. High workforce demand fluctuations of approximately 24% thereby motivated a focus on risk-based approaches rather than mean-risk models, because they capture more clearly the uncertainty, and thereby allow



(a) Mean-CvaR results for various values of α

(b) Mean-EE results for various values of β

Figure 7.12: Cost increases with mean-risk modeling relative to risk-neutral optimization

for smarter risk aversion policies than a combination of expected value and risk measure does. Although in practical applications not all models should have to be analyzed, we also solved model 3 and 5 here for the same range of α and β values to illustrate the effect of this uncertainty on those models. We present the results in Figure 7.12. We find that both mean-risk modeling approaches have very low risk-aversion effects. The mean-risk model 4 (CVaR-based mean-risk) has a small effect for very low values of α, which means, put simply, that this model could be used for optimizing a combination of the average outcomes and a few highly expensive realizations. This concept however might be less intuitive than the corresponding CVaR-based risk approach. The EE-based mean-risk model, in contrast, shows a small risk aversion effect only for small values of β which implies that the uncertainties that the risk term in the objective covers are too small, so that the trade-off between average costs and risks is imbalanced.

The risk-based approach using the CVaR turns out to indeed leave most flexibility to control risks in this real-life warehouse case. However, we also observe a strong sensitivity of the risk aversion on the expected total costs. Already values for α which are slightly smaller than one cause significantly higher cost. A value of $\alpha = 0.95$, for example, which suggests a staffing policy that is optimal for an average of the 95% most expensive scenarios, results in expected total costs that are already 10.58% higher than a risk-neutral optimization.

The warehouse should therefore use an CVaR-based modeling approach when aiming to take risk aversion into account into their staff planning method. However, the strong increase of the expected total costs with decreasing values of α also indicates that risk aversion policies can become very expensive for this company. The reason for this behavior is again the low difference between pre-scheduling and recourse costs. In the risk-neutral approach the expected total costs are minimized which might lead several days in the planning period in which recourse cost are accepted, because they result in the lowest costs. The CVaR model, in contrast, aims to minimize the risk of these recourse costs by over-staffing with slightly less expensive warehouse personnel. This results in a small cost advantage for busy days, but implies high unnecessary costs for the other days.

7.7 Conclusions

In this chapter we considered a staffing problem for commercial warehouses which is often accompanied with high planning uncertainties. We analyzed risk optimization approaches in comparison with an expected value-based optimization approach with the help of multistage stochastic modeling. We developed a decision support tool which can assist to control risks in specific practical warehouse settings and we applied the decision tool in a Dutch commercial warehouse case, for which risk control with the multi-period conditional value at risk appeared to be most applicable among the four approaches.

The contribution of this work is twofold. First, we designed a decision support tool that can be used by practitioners together with an analysis and explanation of the decisions to be made and their impact on momentary outcomes. Testing the tool for a real-life case has shown its applicability. For the case that we studied here the risk modeling approach with the conditional value at risk has provided the best results.

Second, from an academic perspective we make a step further toward the use of stochastic modeling approaches in logistics problems, which has been often recommended by researchers (e.g., Gong and De Koster, 2011; De Koster et al., 2007). Stochastic approaches allow for all kinds of risk management in an optimization and some examples are provided in this work.

The use of risk measures in a logistics contexts has also revealed a distinction compared to the application in financial mathematics. While for financial products uncertainty is usually accompanied with gains on the one hand and losses on the other; here, we observed another effect. For example, an increase of the variation of labor shortage leads to significantly higher costs, even though the average of shortages is the same. Clearly, while the total costs suffer from highly expensive outcomes, they do not profit from exceptionally low shortage realizations, which leads to the observed effect. Risk control in logistics planning problems enables the decision maker to quantify certain risks and can take this effect into account in future research.

Chapter 8

Conclusions and Future Research

Many online retailers are facing the need for changing their operational policies to fulfill customer wishes. Warehouses play in this context a major role since they are responsible for appropriate storage of a large number of goods, for processing of orders within very short response times, for an accurate inventory management, and for appropriate replenishment. All these issues are usually cost intensive and sophisticated operational polices are needed to react on a firm's specific situation. Driven by the need for adjustments of traditional warehouse problems due to product returns, high service quality, and high planning uncertainties, we suggested in this thesis a number of solution methods for logistics decision problems to account for newly arisen phenomena.

This PhD thesis has focused on essential factors and challenges that complicate warehouse operation processes in today's e-commerce environment and presented solution approaches to deal with them. More precisely, this work proposes methods to deal with high product return rates in warehouses as well as for staff planning under uncertainty. All proposed solution methods have been validated by means of numerical experiments and/or were applied to real-life cases which showed promising results for significant savings in warehouse operation costs and an increase in service performance.

In Chapter 3-5 we focused on warehouse order picking problems with respect to the integration of product returns and by accounting for service performance to answer the research questions 1-4 as specified in Section 1.7. Chapter 3 proposes modeling approaches to incorporate service-oriented performance measurement in batching procedures. We provide an example of how consumer-oriented targets can be integrated with efficiency-based objectives of warehouse operations, which addresses research question 1. A real-life case demonstrates that processing times can significantly be reduced with 46% if service performance is incorporated in the optimization objective. Furthermore the proposed batching method is designed to integrate order picking with returns processing. Experiments with the case of a library warehouse (i.e., with a return rate of 100%) showed possible savings of 31% in travel distances when order picking and returns processing are integrated. Chapter 3 thereby provides an answer to research question 2.

Chapter 4 presents a routing method to deal simultaneously with orders and returns and thus answers research question 3. It builds on Chapter 3 by presenting a widely applicable routing algorithm which creates near-optimal routes for batches, which consist of customer order requests to be picked as well as product returns to be brought back to specific storage locations. In both Chapters experiments are reported which show a strong potential to reduce labor and time effort by integrating product returns in the order picking process.

Research question 4, addressing the interrelation of batching and routing, is approached in Chapter 5 in which we presented an integrative approach to combine batch formation and order picker routing. The integration of several warehouse operation problems into joint optimization approaches can result in major advantages if they are interrelated. We propose in Chapter 5 a solution procedure which simultaneously determines batch composition and short routes for each batch. The results of our numerical experiments support the concept of integrating influential warehousing problems by indicating 16.5% potential cost savings for joint in comparison with separated batch and route formation. In total, we proposed solution approaches for both, integrated batch and route formation, for warehouse situations in which both policies have sufficient flexibility, and separated policies, in cases in which one or the other policy is restricted by other warehouse design issues.

Chapters 6 and 7 deal with planning problems under uncertainty in warehouse staffing decisions and thereby address research question 5. The high fluctuations in labor demand of warehouses, the large impact of small disturbances, and high labor costs make warehouse staffing decisions a complex task. Chapter 7 builds on existing stochastic modeling approaches to control the risks of staffing decisions with the help of a decision support tool. This tool provides guidance to choose between a risk-neutral and four forms of risk-averse optimization. Specific uncertainty sources of warehouses are considered as well as regular demand fluctuations and labor shortage. The tool is tested for a real-life case to demonstrate its applicability.

The implementation of the in Chapter 3-5 proposed methods in practical applications requires a careful consideration of the specific warehouse case. While the actual implementation of the solution algorithm requires computer supported order picking systems (e.g., to route order pickers operationally), which are already available in many e-commerce warehouses, the possibility to integrate product returns has to be analyzed in depth. For example, in warehouses in which random storage location assignment is used, the integration of product returns might save time and costs, but it does not require new methods since returns can most likely be delivered at locations which the order picker visits to pick products. Furthermore, it has to be tested how the order picking and returns processing can be integrated operationally, how physical size and weight of the products complicate the problem, and how much preparation and sorting time must be added when, for example, an integrated batching method is used.

8.1 Future research

This thesis motivates several promising directions for future research. Driven by the results of Chapter 3 customer service objectives should play a vital role in warehouse operation designs in future studies. Search and waiting times, for example, have been identified to strongly influence the customer's perception of service quality (Keeney, 1999), so that in future research not only short delivery times, but also just-in-time delivery objectives should be considered. Depending on the warehouse situation the right manner of performance evaluation and multifactorial measures will play a crucial role in warehouse research and practice. Moreover, new indicators of performance have become important in a competitive environment and require research attention when designing new policies. Environmental friendliness, sustainability, and an employer's social responsibility are some examples of relevant indicators that will play an increasing role in the future. The consumers' perception of quality, i.e., the values of those indicators is difficult to measure and hard to balance with cost related investments and outcomes. Future research is asked to find appropriate measurement tools and frameworks to take issues like those into account. Researchers should continue to focus on performance measurement by identifying the relevant features of specific scenarios, finding a way to combine them in performance expressions, and evaluate the measurement tool in comparison with traditional measures and the added value of the newly measured features. Also a framework which categorizes performance indicators is of interest.

Further, this thesis has clearly demonstrated a potential to save costs when interrelated warehouse operation problems are approached jointly. Future research should continue to explore integrated warehouse operation methods to account for the interdependency of different problems. For example, the storage location assignment (see Chapter 2) exerts a substantial impact on order picking performance (Theys et al., 2010), so that an optimal corresponding storage location assignment policy should account for routing options and the implemented batching policy. Staff scheduling problems under uncertainty are interrelated with cross-training policies and error rate minimization problems. The discussion of integrated approaches should furthermore take practical limitations into account. Since warehouse operations problems a usually already mathematically complex when they are considered isolated, optimal approaches with sufficiently short computation times will be challenging to find. In those case metaheuristic search techniques might be promising direction for further research on integrated warehousing problems.

With respect to the integration of warehouse processes, as we have proposed for order picking and returns processing, new challenges arise to optimize the material handling flow before and after the order picking. Batch composition, sorting, and consolidation might require more processing time when orders and returns are processed together. Appropriate batch composition methods and sorting systems are required to maintain the cost savings that are possible when orders and returns are integrated in batching and order picker routing. Since analytical and numerical analyses alone might not be able to reflect the entire complexity of a practical problem, research which is guided by specific practical needs might be the best option to focus on for future research.

Moreover, although more calculation time can be allowed for the meta-heuristic approach in Chapter 5, because it determines a schedule for an entire order picking shift, still, unexpected, short-notice rescheduling might be necessary sometimes. This requires solution approaches with shorter computation times which in turn allow for more planning flexibility. In that respect also further research on the in Chapter 5 considered integrated order picking problem is of interest to explore lower bounds, which can facilitate the design of more powerful solution techniques.

Future research should also focus on stochastic methods to design warehouse policies that are affected by uncertainties. Labor scheduling is only one problem that usually requires longer planning horizons than the available exact information on future demand would allow. Availability of products and resources, lateness, and technical failure might be other examples of uncertainties that complicate planning problems.

Last, the decision support tool, presented in Chapter 7, could be extended by incorporating other suitable risk measures, a more detailed decision guideline design, or other uncertainty sources. The outcomes of our analyses revealed a relatively general directive to determine a risk modeling approach, which suggests a potential extension of the research design to other staff scheduling contexts and other logistics problems that are influenced by uncertain factors. Future research is required to test the applicability of our decision tool for similar problems.

List of Figures

List of Tables

References

Abernathy, W. J., Baloff, N., Hershey, J. C., and Wandel, S. (1973). A three-stage manpower planning and scheduling model - a service-sector example. *Operations Research*, 21(3):693–711.

Acerbi, C. and Tasche, D. (2002). Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes*, 31(2):379–388.

Ahn, T., Ryu, S., and Han, I. (2004). The impact of the online and offline features on the user acceptance of Internet shopping malls. *Electronic Commerce Research and Applications*, 3(4):405–420.

Ai, T. J. and Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, 36(5):1693–1702.

Aksin, Z., Armony, M., and Mehrotra, V. (2007). The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management*, 16(6):665–688.

Albareda-Sambola, M., Alonso-Ayuso, A., Molina, E., and De Blas, C. S. (2009). Variable neighborhood search for order batching in a warehouse. *Asia-Pacific Journal of Operational Research*, 26(5):655–683.

Armstrong, R. D., Cook, W. D., and Saipe, A. L. (1979). Optimal batching in a semi-automated order picking system. *The Journal of the Operational Research Society*, 30(8):711–720.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

Axsäter, S. (1993). Optimization of order-up-to-s policies in two-echelon inventory systems with periodic review. *Naval Research Logistics (NRL)*, 40(2):245–253.

Baker, P. and Halim, Z. (2007). An exploration of warehouse automation implementations: cost, service and flexibility issues. *Supply Chain Management: An International Journal*, 12(2):129–138.

Bard, J. F., Morton, D. P., and Wang, Y. M. (2007). Workforce planning at usps mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1):51–78.

Barnhart, C., M.Cohn, A., Johnson, E. L., Klabjan, D., Nemhauser, G. L., and Vance, P. H. (2003). Airline crew scheduling. In Hall, R. W., editor, *Handbook of Transportation Science*, volume 56 of *International Series in Operations Research & Management Science*, pages 517–560. Springer US.

Bernon, M., Upperton, J., Bastl, M., and Cullen, J. (2013). An exploration of supply chain integration in the retail product returns process. *International Journal of Physical Distribution & Logistics Management*, 43(7):586–608.

Bianchessi, N. and Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 34(2):578–594.

Bonifield, C., Cole, C., and Schultz, R. L. (2010). Product returns on the Internet: A case of mixed signals? *Journal of Business Research*, 63(9–10):1058–1065.

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

Brusco, M. J. (2008). An exact algorithm for a workforce allocation problem with application to an analysis of cross-training policies. *IIE Transactions*, 40(5):495–508.

Brynjolfsson, E., Hu, Y. J., and Smith, M. D. (2003). Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science*, 49(11):1580–1596.

Burt, S. and Sparks, L. (2003). E-commerce and the retail process: a review. *Journal of Retailing and Consumer Services*, 10(5):275–286.

Cachon, G. P. (2001). Exact evaluation of batch-ordering inventory policies in two-echelon supply chains with periodic review. *Operations Research*, 79(1):79–98.

Caruana, A. and Ewing, M. T. (2010). How corporate reputation, quality, and value influence online loyalty. *Journal of Business Research*, 63(9–10):1103–1110.

Chan, F. T. and Chan, H. (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications*, 38(3):2686–2700.

Chang, M. K., Cheung, W., and Tang, M. (2013). Building trust online: Interactions among trust building mechanisms. *Information & Management (published online)*.

CHEMS Consulting (2005). Optimising storage and access in UK research libraries a study for CURL and the british library.

Chen, C.-M., Gong, Y., 'De Koster, R. B., and Van Nunen, J. A. (2010). A flexible evaluative framework for order picking systems. *Production and Operations Management Society*, 19(1):70–82.

Chen, M.-C., Huang, C.-L., Chen, K.-Y., and Wu, H.-P. (2005). Aggregation of orders in distribution centers using data mining. *Expert Systems with Applications*, 28(3):453–460.

Chen, M.-C. and Wu, H.-P. (2005). An association-based clustering approach to order batching considering customer demand patterns. *Omega*, 33(4):333–343.

Chepesiuk, R. (1999). Reaching critical mass: Off-site storage in the digital age. *American Libraries*, 30(4):40–43.

Chiang, W. K. and Monahan, G. E. (2005). Managing inventories in a two-echelon dual-channel supply chain. *European Journal of Operational Research*, 162(2):325–341.

Cox, J. and Dale, B. (2001). Service quality and e-commerce: an exploratory analysis. *Managing Service Quality*, 11(2):121–131.

De Koster, M. B. M., Van Der Poort, E. S., and Wolters, M. (1999). Efficient orderbatching methods in warehouses. *International Journal of Production Research*, 37(7):1479–1504.

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

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

De Koster, R. B., De Brito, M. P., and Van De Vendel, M. A. (2002). Return handling: An exploratory study with nine retailer warehouses. *International Journal of Retail & Distribution Management*, 30(8):407–421.

Dekker, R., De Koster, M. B. M., Roodbergen, K. J., and Van Kalleveen, H. (2004). Improving order-picking response time at Ankor's warehouse. *Interfaces*, 34(4):303–313.

Díaz, B. A., González-Torre, P., and García, V. (2002). A capacity management model in service industries. *International Journal of Service Industry Management*, 13(3):286–302.

Doherty, N. F. and Ellis-Chadwick, F. (2010). Internet retailing: The past, the present and the future. *International Journal of Retail & Distribution Management*, 38(11/12):943–965.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation*, 1(1):53–66.

Dunbar, M., Froyland, G., and Wu, C.-L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Science*, 46(2):204–216.

Ehrgott, M. and Ryan, D. M. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150.

Elsayed, E. A., Lee, M.-K., Kim, S., and Scherer, E. (1993). Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals. *International Journal of Production Research*, 31(3):727–738.

Elsayed, E. A. and Stern, R. (1983). Computerized algorithms for order processing in automated warehousing systems. *International Journal of Production Research*, 21(4):579–586.

Ernst, A., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27.

Eveborn, P., Flisberg, P., and Rönnqvist, M. (2006). Laps carean operational system for staff planning of home care. *European Journal of Operational Research*, 171(3):962–976.

Eveborn, P. and Rönnqvist, M. (2004). Scheduler - a system for staff planning. *Annals of Operations Research*, 128:21–45.

Fang, C. and Wang, L. (2012). An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Computers & Operations Research*, 39(5):890–901.

Fowler, J. W., Wirojanagud, P., and Gel, E. S. (2008). Heuristics for workforce planning with worker differences. *European Journal of Operational Research*, 190(3):724–740.

Fragnière, E., Gondzio, J., and Yang, X. (2010). Operations risk management by optimally planning the qualified workforce capacity. *European Journal of Operational Research*, 202(2):518–527.

Gademann, A., Van Den Berg, J., and Van Der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE Transactions*, 33(5):385–398.

Gademann, N. and Van De Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*, 37(1):63–75.

Gefen, D. (2000). E-commerce: The role of familiarity and trust. *Omega*, 28(6):725–737.

Goksal, F. P., Karaoglan, I., and Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 65(1):39–53.

Gong, Y. and De Koster, M. (2009). Approximate optimal order batch sizes in a parallel aisle warehouse. In *Lecture notes in economics and mathematical systems*, volume 619, pages 175–194. L. Bertazzi and M. Grazia Speranza and J.A.E.E. Nunen (editors).

Gong, Y. and De Koster, R. B. M. (2011). A review on stochastic models and analysis of warehouse operations. *Logistics Research*, 3(4):191–205.

Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21.

Gunasekaran, A., Patel, C., and Tirtiroglu, E. (2001). Performance measures and metrics in a supply chain environment. *International Journal of Operations & Production Management*, 21(1/2):71–87.

Ha, H. and McGregor, . S. L. T. (2013). Role of consumer associations in the governance of e-commerce consumer protection. *Journal of Internet Commerce*, 12(1):1–25. published online.

Hamberg, R. and Verriet, J. (2012). *Automation in warehouse development*. Springer.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449 – 467.

Hansen, P. and Mladenović, N. (2003). Variable neighborhood search. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 145–184. Springer US.

Hassan, M. M. D. (2002). A framework for the design of warehouse layout. *Facilities*, 20(13/14):432–440.

Heikkilä, J. (2002). From supply to demand chain management: Efficiency and customer satisfaction. *Journal of Operartions Management*, 20(6):747–767.

Heinze, T. (2008). *Ein Verfahren zur Dekomposition mehrstufiger stochastischer Optimierungsprobleme mit Ganzzahligkeit und Risikoaversion*. PhD thesis.

Heitsch, H. and Römisch, W. (2009). Scenario tree moedling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406.

Henn, S., Koch, S., Doerner, K. F., Strauss, C., and Wäscher, G. (2010). Metaheuristics for the order batching problem in manual order picking systems. *BuR - Business Research*, 3(1):82–105.

Heragu, S. S. (2008). *Facilities Design*. Taylor and Francis Group, third edition.

Hernández-Pérez, H. and Salazar-González, J.-J. (2004). A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, 145(1):126–139.

Ho, Y.-C., Su, T.-S., and Shi, Z.-B. (2008). Order-batching methods for an order-picking warehouse with two cross aisles. *Computers and Industrial Engineering*, 55(2):321–347.

Ho, Y.-C. and Tseng, Y.-Y. (2006). A study on order-batching methods of order-picking in a distribution centre with two cross-aisles. *International Journal of Production Research*, 44(17):3391–3417.

Hodgson, T. J. and Lowe, T. J. (1982). Production lot sizing with material-handling cost considerations. *A I I E Transactions*, 14(1):44–51.

Hoff, A., a, I. G., Laporte, G., and Løkketangen, A. (2009). Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 192:755–766.

Hong, T.-P., Wang, H.-S., and Chen, W.-C. (2000). Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, 6(4):439–455.

Hsieh, L.-F. and Huang, Y.-C. (2011). New batch construction heuristics to optimise the performance of order picking systems. *International Journal of Production Economics*, 131(2):613–630.

Hsu, C.-M., Chen, K.-Y., and Chen, M.-C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, 56(2):169–178.

Hwang, H., Baek, W., and Lee, M.-K. (1988). Clustering algortihms for order picking in an automated storage and retrieval system. *International Journal of Production Research*, 26(2):189–201.

Hwang, H. and Kim, D. G. (2005). Order-batching heuristics based on cluster analysis in a low-level picker-to-part warehousing system. *International Journal of Production Research*, 43(17):3657–3670.

Jeang, A. (1994). Flexible nursing staff planning when patient demands are uncertain. *Journal of Medical Systems*, 18(3):125–138.

Kazarlis, S. A., Bakirtzis, A. G., and Petridis, V. (1996). A genetic algorithm solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 11(1):83–92.

Keeney, R. L. (1999). The value of Internet e-commerce to the customer. *Management Science*, 45(4):533–541.

Kim, B.-I., Heragu, S. S., Graves, R. J., and St. Onge, A. (2003). Realization of a short cycle time in warehouse replenishment and order picking. *International Journal of Production Research*, 41(2):349–364.

Kohl, D. F. (2003). Paper and digital repositories in the united states. *Liber Quarterly*, 13(3/4):241–253.

Kuo, Y.-F., Wu, C.-M., and Deng, W.-J. (2009). The relationships among service quality, perceived value, customer satisfaction, and post-purchase intention in mobile value-added services. *Computers in Human Behavior*, 25(4):887–896.

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

Lee, M.-K. and Elsayed, E. A. (2005). Optimization of warehouse storage capacity under a dedicated storage policy. *International Journal of Production Research*, 43(9):1785–1805.

Li, H. and Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows.

Liao, S., Koole, G., Van Delft, C., and Jouini, O. (2012). Staffing a call center with uncertain non-stationary arrival rate and flexibility. *OR Spectrum*, 34:691–721.

Malmborg, C. J. (1996). An integrated storage system evaluation model. *Applied Mathematical Modelling*, 20(5):359–370.

Matusiak, M., De Koster, R., Kroon, L., and Saarinen, J. (2013). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research (published online)*.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377 – 386.

Mollenkopfa, D. A., Frankel, R., and Russo, I. (2011). Creating value through returns management: Exploring the marketingoperations interface. *Journal of Operations Management*, 29(5):391–403.

Montané, F. A. T. and Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research*, 33(3):595–619.

Mosheiov, G. (1994). The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research*, 79(2):299–310.

Mostard, J., De Koster, R., and Teunter, R. (2005). The distribution-free newsboy problem with resalable returns. *International Journal of Production Economics*, 97(3):329–342.

Mostard, J. and Teunter, R. (2006). The newsboy problem with resalable returns: A single period model and case study. *European Journal of Operational Research*, 169(1):81–96.

Nagy, G. and Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162(1):126–141.

Nitecki, D. A. and Kendrick, C. L. (2001). *Library Off-Site Shelving*. Libraries Unlimited Inc.

Ofek, E., Katona, Z., and Sarvary, M. (2011). ”bricks and clicks”: The impact of product returns on the strategies of multichannel retailers. *Marketing Science*, 30(1).

Önüt, S., Tuzkaya, U. R., and Doğaç, B. (2008). A particle swarm optimization algorithm for the multiple-level warehouse layout design problem. *Computers & Industrial Engineering*, 54(4):783–799.

Pan, C.-H. and Liu, S.-Y. (1995). A comparative study of order batching algorithms. *International Journal of Management Science*, 23(6):691–700.

Papadimitriou, C. H. (1977). The euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244.

Pechtl, H. (2003). Adoption of online shopping by german grocery shoppers. *The International Review of Retail, Distribution and Consumer Research*, 13(2):145–159.

Petersen, C. G. (1995). Routing and storage policy interaction in order picking. pages 1614–1616.

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

Petersen, C. G. and Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92(1):11–19.

Ramanathan, R. (2011). An empirical analysis on the influence of risk on relationships between handling of product returns and customer loyalty in E-commerce. *International Journal of Production Economics*, 130(2):255–261.

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

Ribeiro, G. M. and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735.

Rockafellar, R. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471.

Roodbergen, K. J. and De Koster, R. (2001a). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883.

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

Roodbergen, K. J. and Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194(2):343–362.

Roodbergen, K. J. and Vis, I. F. A. (2006). A model for warehouse layout. *IIE Transactions*, 38(10):799–811.

Roubos, A. (2012). *Service - Level Variability and Impatience in Call Centers*. PhD thesis.

Rouwenhorst, B., Reuter, B., Stockrahm, V., Van Houtum, G., Mantel, R. J., and Zijm, W. H. M. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, 122(3):515–533.

Ruben, R. A. and Jacobs, F. R. (1999). Batch construction heuristics and storage assignment strategies for walk/ride and pick systems. *Management Science*, 45(4):575–596.

Sadjadi, S. J., Soltani, R., Izadkhah, M., Saberian, F., and Darayi, M. (2011). A new nonlinear stochastic staff scheduling model. *Scientia Iranica*, 18(3):699–710.

Schaefer, A. J., Johnson, E. L., Kleywegt, A. J., and Nemhauser, G. L. (2005). Airline crew scheduling under uncertainty. *Transportation Science*, 39(3):340–348.

Seaman, S. (2004). High-density off-site storage in north american research libraries.

Shulman, J. D., Coughlan, A. T., and Savaskan, R. C. (2010). Optimal reverse channel structure for consumer product returns. *Marketing Science*, 29(6):1071–1085.

Stenger, A., Vigo, D., Enz, S., and Schwind, M. (2013). An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47(1):64–80.

Stock, J. R. and Mulki, J. P. (2009). Product returns processing: A examination of practices of manufacturers, wholesalers/distributors, and retailers. *Journal of Business Logistics*, 30(1):33–62.

Su, X. (2009). Consumer return policies and supply chain performance. *Manufacturing & Service Operations Management*, 11(4):595–612.

Tagaras, G. and Vlachos, D. (2001). A periodic review inventory system with emergency replenishments. *Management Science*, 47(3):415–429.

Tang, C. S. (2010). A review of marketing-operations interface models: From co-existence to coordination and collaboration. *International Journal of Production Economics*, 125(1):22–40.

Tharmmaphornphilas, W. and Norman, B. A. (2007). A methodology to create robust job rotation schedules. *Annals of Operations Research*, 155(1):339–360.

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

Tompkins, J. A., White, J. A., Bozer, Y. A., and Tanchoco, J. (2010). *Facilities Planning*. Don Fowley, fourth edition.

Tsai, C.-Y., Liou, J. J. H., and Huang, T.-M. (2008). Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time. *International Journal of Production Research*, 46(22):6533–6555.

Van Landeghem, H. and Vanmaele, H. (2002). Robust planning: A new paradigm for demand chain planning. *Journal of Operations Management*, 20(6):769–783.

Vaughan, T. S. and Petersen, C. G. (1999). The effect of warehouse cross aisles on order picking efficiency. *International Journal of Production Research*, 37(4):881–897.

Wilson, H. G. (1977). Order quantity, product popularity, and the location of stock in a warehouse. *A I I E Transactions*, 9(3):230–237.

Won, J. and Olafsson, S. (2012). Joint order batching and order picking in warehouse operations. *International Journal of Production Research*, 43(7):1427–1442.

Wruck, S., Roodbergen, K. J., Veenstra, M., and Dijkstra, A. (2013a). Order picker routing in warehouses with pickup and delivery. *under review at the European Journal of Operational Research*.

Wruck, S., Vis, I. F. A., and Boter, J. (2013b). A decision tool for risk control in warehouse staff planning. *(work in progress)*.

Wruck, S., Vis, I. F. A., and Boter, J. (2013c). An iterated local search approach for integrated batching and routing in warehouses with orders and returns. *under review at IIE Transactions*.

Wruck, S., Vis, I. F. A., and Boter, J. (2013d). Time-restricted batching models and solution approaches for integrated forward and return product flow handling in warehouses. *Journal of the Operational Research Society*, 64(10):1505–1516.

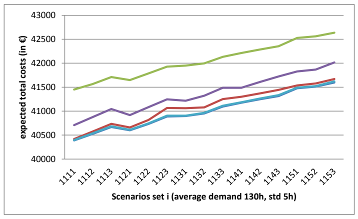
Xu, P. J., Allgor, R., and Graves, S. C. (2009). Benefits of reevaluating real-time order fulfilment decisions. *Manufacturing & Service Operations Management*, 11(2):340–355.

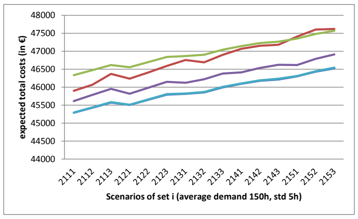
Zhang, X., Chakravarthy, A., and Gu, Q. (2009). Equipment scheduling problem under disruptions in mail processing and distribution centres. *Journal of the Operational Research Society*, 60(5):598 –610.

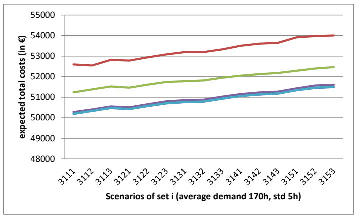
Zhou, Y.-W. (2003). A multi-warehouse inventory model for items with time-varying demand and shortages. *Computers & Operations Research*, 30(14):2115–2134.

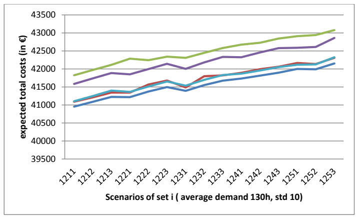
Appendix A

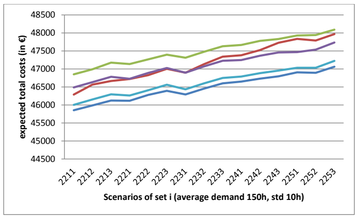
Detailed figures of results of scenario set i

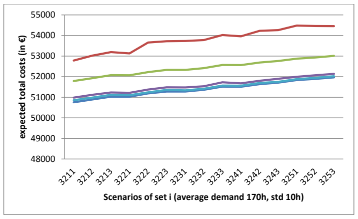


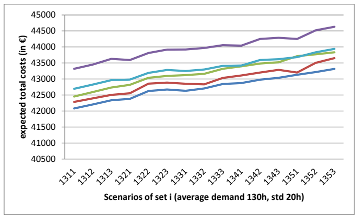


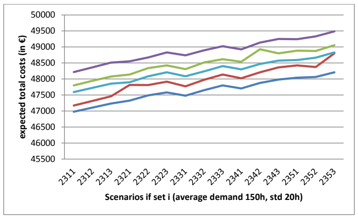


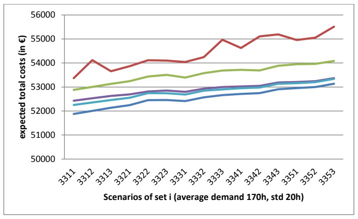


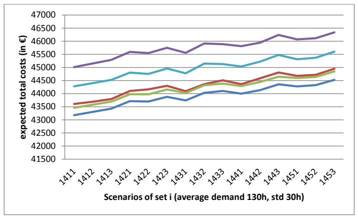


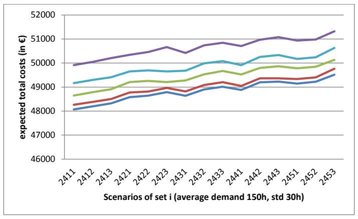


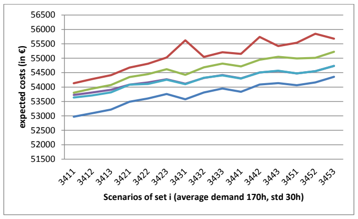


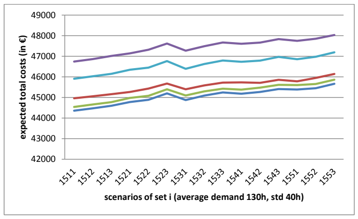


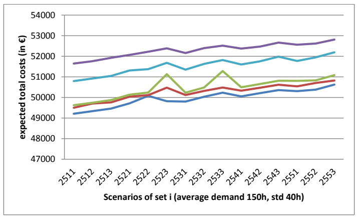


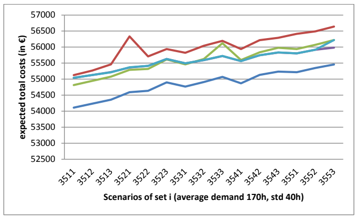












🟦 Model 1
🟥 Model 2
🟩 Model 3
🟪 Model 4
🟦 Model 5

Chapter 9

Summary

English Summary

In today's e-commerce continuous technological development, high competition, and especially high service standard requirements pose major challenges for the logistics operations of a company. Especially warehouses, which are responsible for the storage of goods, which are picked, packed, and shipped in response to customer orders, are requested to adjust their operations due to recent developments. High product return rates are one of those phenomena that complicate logistics operations. Significant return rates have become common for many companies and can, depending on the product category, be very high. For consumers the opportunity to shop online is usually accompanied with very liberal return policies, which induces those high return rates. Returned products in warehouses need to be unpacked, inspected, and re-integrated in the warehouse stock before they can be resold. With an increasing number of returns the processing becomes labor intensive so that the need for well-performing methods to deal with returns is prominent. This motivates reconsideration and redesign of traditional logistics processes which facilitate logistics efficiency on the one hand, as well as quick and reliable deliveries on the other hand.

The first chapters of this dissertation deal with order picking operation problems in warehouses of companies which face high return rates. Order picking operations contribute to more than 55% to the overall operation costs of warehouses. We show that when product returns arrive at the warehouse next to customer orders, there is a potential to save labor costs by integrating the two flows in some warehouse processes. Order batching and order picker routing are two warehouse problems that apply to both order picking and returns processing, which suggests to integrate the product flows. With batching a number of requests is grouped

into smaller sets each of which to be picked in one route, which in turn facilitates an efficient order picking process. If those requests consist of not only customer orders but also product returns, larger batches can be formed than the transport capacity would actually allow, if only the batch consists of the right mix of orders and returns. In Chapter 3 we considered the problem of batching a number of customer orders together with product returns in order to achieve quick order picking, while orders and returns are processed together in order to save time and costs. Order picker routing, which we discussed in Chapter 4, allows for an integration in a similar manner. Once a batch of orders and returns has been found, the order picker must be routed not only with the shortest possible route, but also under consideration of the transport capacity. We presented a well-performing method to determine such routes and also studied the impact that a specific mix of orders and returns in batches has on the quality of solutions.

In Chapter 5 we discussed the opportunity to consider the order batching and order picker routing problem holistically. Obviously, the two problems are interrelated and the performance of one method depends on the performance of the other. While sticking to context of integrated order and return flows, we proposed a model to also integrate batching and routing and presented a solution method to solve large-scale problems.

In Chapter 6 and 7, we focused on another important driver of warehouse costs, namely staff planning. We studied a staff planning problem for warehouses in situations of high workforce demand fluctuations, which are - just like product returns - a common issue and problematic for e-commerce retailers. Labor is usually highly cost-intensive and has to be used most efficiently by optimal scheduling. We discussed five optimization approaches to incorporate risk management in a staff schedule optimization problem and presented a decision tool for warehouse managers which can assist them in deriving low-cost solutions by simultaneously controlling risks of shortages.

Overall we addressed in this thesis a number of problems, which arise in many warehouses of e-commerce retailers and presented suited solution approaches for those problems. The performance of our designed algorithms we demonstrated by means of numerical experiments and partially by considering real-life situations. Our experiments showed that significant cost savings can be achieved by integrating incoming and outgoing product flows in warehouses, as well as by combining and jointly solving interrelated warehouse problems.

Samenvatting

Magazijnprocessen opnieuw bezien - nieuwe uitdagingen en methoden

Niet alleen de enorme groei van e-commerce, maar ook de hevige concurrentie, de doorlopende ontwikkelingen en de stijgende servicestandaarden stellen de logistieke operaties van retailers voor grote uitdagingen. Retailketens worden steeds complexer en klanten worden steeds veeleisender. Dat betekent dat een efficiënte operatie, goede leverprestaties en een flinke dosis innovativiteit essentiële voorwaarden zijn voor succes. Met name in de magazijnen die verantwoordelijk zijn voor het opslaan, verzamelen, verpakken en verzenden van producten, is de druk groot.

Een belangrijke complicerende factor vormen de retourstromen. Veel online retailers hanteren een tolerant retourbeleid om de service te vergroten en op die manier het vertrouwen van klanten te winnen. Het resultaat is dat afhankelijk van de productgroep tot wel 75 procent van de verkochte producten wordt teruggestuurd. In hun magazijnen dienen retailers al deze producten uit te pakken, te controleren en weer op voorraad te leggen voordat ze opnieuw in de verkoop kunnen. Omdat het aantal retouren toeneemt, wordt dit proces steeds arbeidsintensiever. Daardoor groeit de behoefte aan efficiënte methoden voor het afhandelen van retouren. Reden genoeg om de traditionele logistieke processen opnieuw tegen het licht te houden en opnieuw te ontwerpen. Alleen dan kan een efficiënte logistiek inclusief adequate retourprocessen en snelle en betrouwbare leveringen ontstaan.

In deze dissertatie worden methodieken voorgesteld voor het integreren van de retourprocessen in de bestaande orderverzamelprocessen. Integratie in orderverzamelprocessen, die gemiddeld 55 procent van de totale operationele kosten uitmaken, leidt tot lagere operationele kosten. In de eerste vier hoofdstukken na de introductie staan daarom de orderverzamelprocessen centraal. Het verzamelen van orders en het verwerken van retouren worden samengebracht in twee processen: het samenstellen van groepen van orders (batches) en het bepalen van de looproute van medewerkers langs de opslaglocaties. Het samenstellen van batches betekent dat meerdere klantorders of retouren worden samengevoegd tot n grote opdracht die aan een enkele medewerker kan worden toegewezen. De looproute door het warehouse bepaalt in belangrijke mate de efficiëntie van medewerkers bij het verzamelen of juist weer op voorraad leggen van producten.

In veel magazijen worden orders gegroepeerd om de efficiëntie van het orderverzamelproces te vergroten en meer orders in kortere tijd te kunnen afhandelen. De grootte van de batch wordt bepaald door de beperkte capaciteit die de medewerker heeft om alle verzamelde producten op zijn ronde mee te nemen. Als we nu het verzamelen van orders en het weer op voorraad leggen van geretourneerde producten combineren, kunnen we batches samenstellen die groter zijn dan de transportcapaciteit van de medewerker toelaat. Met andere woorden: medewerkers starten hun route door het magazijn met een kar vol geretourneerde producten die weer op voorraad moeten worden gelegd en eindigen met een kar vol producten die door klanten zijn besteld. Dat is mogelijk zolang de juiste mix van orders en retouren wordt samengebracht. In deze dissertatie tonen we aan de hand van een casus in een bibliotheekmagazijn aan dat een dergelijke integratie van orders en retouren tijd en kosten bespaart zonder dat dat ten koste gaat van de responstijden en de kwaliteit van de dienstverlening aan klanten. Om de prestaties van een gentegreerd orderverzamel- en retourproces te kunnen meten, zijn prestatie-indicatoren gedefinieerd waarmee zowel de efficiëntie van de operatie als de kwaliteit van de dienstverlening worden gemeten.

Na het samenstellen van batches komt in deze dissertatie het bepalen van looproutes aan bod. Het is de uitdaging om gegeven een batch die uit zowel orders als retouren bestaat - de kortst mogelijke route te berekenen langs de locaties in het magazijn die de medewerker moet bezoeken. Voor dat doel is een genetisch algoritme ontwikkeld dat het mogelijk maakt om in zon kort mogelijke tijd de optimale route voor grote batches te berekenen. Het algoritme is getest middels een aantal numerieke experimenten. Tevens is onderzocht hoe orders en retouren zo efficiënt mogelijk tot batches kunnen worden gecombineerd, terwijl de medewerker voldoende flexibilteit houdt bij het sorteren van producten op de kar.

Deze twee hoofdstukken over batching en routering worden gevolgd door een hoofdstuk waarin we onderzoeken of het mogelijk is om het formeren van batches en bepalen van looproutes gelijktijdig uit te voeren. Daaruit blijkt dat het soms onmogelijk is om beide activiteiten te integreren, bijvoorbeeld door de grote complexiteit van elk probleem afzonderlijk of het gebrek aan flexibiliteit in de keuze van een batch- of routeringsmethode. Als het echter mogelijk is om beide activiteiten wel te integreren en het probleem holistisch te benaderen, kan dat zowel tijd als kosten besparen. In deze dissertatie wordt hiervoor een oplossing gepresenteerd en gevalideerd, die gebaseerd is op een iteratieve locale zoekmethode en verscheidene neighborhood structuren.

Het laatste deel van deze dissertatie gaat over een andere belangrijke uitdaging in logistieke operaties: personeelsplanning. Arbeid is duur, zodat het belangrijk is om de beschikbare arbeidskrachten met optimale planningstechnieken zo efficiënt mogelijk in te zetten. Online retailers kampen bovendien met grote onzekerheden in het planningsproces, aangezien de vraag uit de markt per definitie onbekend is en sterk kan fluctueren. Om daarmee adequaat om te gaan is risico management vereist. In deze dissertatie bespreken wij vijf methoden om risico management in personeelsplanning te incorpororeren. Daarnaast presenteren we een instrument waarmee magazijn managers een kostenefficiënte personeelsplanning kunnen maken en gelijktijdig de kansen op personeelstekorten kunnen minimaliseren.

Deze dissertatie levert een bijdrage aan de wetenschappelijke literatuur door een aantal klassieke problemen voor e-commerce warehouses opnieuw aan een onderzoek te onderwerpen. Nieuwe methodes voor de afhandeling van retouren en het omgaan met de sterk fluctuerende vraag leidt tot nieuwe mogelijkheden voor een efficiëntere logistiek, die we in deze publicatie bespreken.