Technische Universität München

Lehrstuhl für Logistik und Supply Chain Management

Inventory optimization under uncertainty and fixed cost structures: Theoretical foundations and data-driven applications

Dipl.-Kfm. Univ. Florian Niklas Taube

Vollständiger Abdruck der von der Fakultät für Wirtschaftswissenschaften der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Martin Grunow
Prüfer der Dissertation: :	1.	Prof. Dr. Stefan Minner
	2.	Prof. Nesim K. Erkip, Ph.D. Bilkent University Ankara / Türkei

Die Dissertation wurde am 16.11.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15.12.2017 angenommen.

Acknowledgements

It is not possible to write a doctoral thesis without the continuous help of others.

My supervisor Prof. Dr. Stefan Minner has always been the strongest support during my time as a doctoral candidate. Due to a plethora of ideas and his far-reaching knowledge on a broad range of topics, he constantly sparked my enthusiasm for research. I am very grateful that I was given the opportunity to work on a huge variety of topics in inventory control, production and transportation, which has greatly broadened my understanding. I also want to thank Prof. Nesim Erkip for taking time to review my dissertation and being part of the examination committee and Prof. Dr. Martin Grunow for being the chairman of said committee.

During my work at the chair of Logistics and Supply Chain Management at the Technical University of Munich I was able to get to know lots of interesting and always helpful people who I am proud to call friends and all of whom I want to thank deeply: Yuka Akasaka, Szymon Albinski, Dr. Christian Bohner, Dr. Maximilian Budde, Dr. Pirmin Fontaine, Dr. Miray Közen, Sebastian Malicki, Thitinan "Kate" Pholsook, Jun.-Prof. Dr. Anna-Lena Sachs, Dr. Martin Stößlein and Dariush Tavaghof-Gigloo. Special thanks go to Christian Mandl for being my roommate and discussion partner on all scientific and non-scientific topics and for his evening distractions, when needed. And, of course, I want to thank Evelyn Gemkow, who proofread my work and helped, especially, with all the little things pertaining life at the chair. Many thanks also to Özgün Turgut for his collaboration in one of the papers.

I am most grateful to have a wonderful family, who have always supported me. I would like to thank my parents, who taught me to learn, stay humble and live a happy life and my brother Julian, not only for his technical support. My deepest appreciation goes to my wife Wen, who constantly renews my love in her and who has made my life complete: 我爱你. Lastly, I want to thank my little pearl of wisdom Sophia, who teaches me never to take anything too seriously.

Abstract

In the age of growing data availability, advanced decision support is needed to make use of Big Data for practical applications. This thesis contributes to this need and provides insights into cyclical fixed order cost structures, advanced delivery concepts in retail considering handling efforts and an innovative organizational approach to just-in-sequence production.

Current theory describes the optimal inventory policy under stochastic demand and fixed order costs that are non-increasing over time. We consider that fixed costs vary cyclically over time and prove the optimal inventory policy. This policy is complex and differs from the established theory. Yet, we provide nearly optimal simple-to-use policies. We reveal that a forward-buying effect, which aims at reducing the probability to order in high-cost periods, drives the policy structure.

We next consider the retail environment and its handling efforts in the supply chain. Orders incur fixed costs, hence it is more cost efficient to order several products simultaneously. We optimize parameters of inventory control policies that focus on fixed delivery rhythms or coordination of orders across products. We establish data-driven mixed integer linear programs that use historical demand values to model uncertainty. Our approach of setting the policy parameters performs best when lots of data is used and fixed costs are significant. Cost improvements are large when historical data, rather than expected demands, are included. Those cost benefits are driven by a reduction of fixed order and out-of-stock costs. We also show that overflow inventory that does not fit on the shelf significantly drives costs.

In another practical problem, a just-in-sequence module supplier in the automotive industry uses a mixed-model assembly line. Our idea of resequencing the original sequence of the customer improves the production flow at the supplier. We establish a mixed integer linear program that creates an improved sequence and incorporates restoration via a first-in-first-out approach by using mix banks. We find that, with two mix banks, already most of the savings are realized and sequences of a certain length and complexity are most attractive for the approach.

Contents

List of Tables

List of Figures

Acronyms

ANOVA	Analysis of Variance.
AP	Assignment Problem.
BRE	Backroom Effect.
c.d.f.	Cumulative Distribution Function.
CB	Color Batching.
CV	Coefficient of Variation.
DPS	Product Store Decomposition.
DTW	Time Horizon Based Decomposition.
EDM	Expected Demand Model.
FIFO	First-In-First-Out.
FM	Full Model.
GA	Genetic Algorithm.
GAM	Genetic Algorithm Model.
INT	Integrated Mixed Integer Linear Program.
IRP	Inventory Routing Problem.
JIS	Just-In-Sequence.
JIT	Just-In-Time.
JRP	Joint Replenishment Problem.
JRPDD	Joint Replenishment Problem under Dynamic Demand.

LB	Load Balancing.
LL	Limited Lookahead Approach.
MILP	Mixed Integer Linear Program.
ML	Material Leveling.
MSR	Ratio between Major and Minor Set-up Cost.
No-FIFO	Model without FIFO Restriction.
NoBRE	Model without Backroom Effect.
NoDD	Model without Data-driven Approach.
NoJRP	Model without Joint Replenishment.
OEM	Original Equipment Manufacturer.
OOS	Out-Of-Stock.
ORR	Ratio between Order and Shelf Stacking Costs.
PVRP	Periodic Vehicle Routing Problem.
SAVS	Savings Heuristic.
SEQ	Sequential Model.
SGA	Self-Adaptive Genetic Algorithm.
TBO	Time Between Orders.
TSP	Traveling Salesman Problem.
VRP	Vehicle Routing Problem.

Chapter 1.

Introduction

1.1. Motivation

The digital reinvention of industries is unstoppable but far from finished. The biggest improvements in revenues and earnings can be gained by digitizing supply chains (Bughin et al., 2017). To ensure the vision of Industry 4.0 for digital supply chains, one key element is the implementation of prescriptive supply chain analytics (Schrauf & Berttram, 2016). Tools for this advanced decision support make use of the available data not only to anticipate future events but also to provide optimized solutions on how to tackle those events. This enriched information is then given to managers as decision support or decisions are automatically executed (e.g. as in automated order management systems). In the study *Digital Supply Chains: Increasingly Critical for Competitive Edge* (2015), business consultants company AT Kearney and business school WHU asked supply chain managers of leading European companies on their opinion about the impacts of digitalization on their supply chain. The study shows that the majority of respondents see a clear or very strong impact of digitization on increased just-in-time (JIT) sourcing and the reduction of order-to-stock. Positive impacts are mostly seen for better overall supply chain management decisions, increased flexibility and lower inventory and warehousing costs.

In this thesis, we pick up on all those effects of digitization and explicitly use prescriptive analytics to provide decision support, increase flexibility in the just-in-sequence production environment and reduce inventory and warehousing costs in retail.

Industry-specific digitization strategies follow the same direction as above. Retail managers call for heavy analytical approaches down to the item and store level and accompanying innovative store replenishment and delivery processes to be able to cope with omni-channel retail requirements in demand planning (Gibson et al., 2016). For automotive suppliers, big data is generally one of the most relevant digitization trends and intelligent algorithms for decision making in the form of data analytics are a key element when it comes to achieving higher utilization of plant and machinery and smaller production times (Brauchle et al., 2015).

In addition to these technological opportunities, markets have to deal with ever more uncertainty and cost pressure. The stationary retail market is steadily transformed by the rising influence of e-commerce, which, according to the *Global Sector Report - Retail* of Euler Hermes Economic Research (Allouche, 2017), might well account for 15 % of worldwide retail sales by 2020. One key factor that might keep traditional retailers in play is the focus on more efficient logistics (Sharma, 2017). At the same time, profitability is decreasing (Allouche, 2017), which is a situation that will likely not change as disruptive changes like the Amazon takeover and subsequent price drop of Wholefoods happen (Rigby, 2017). For automotive suppliers, the *Global Automotive Supplier Study 2016* by Roland Berger / LAZARD (Berret et al., 2016) emphasizes that there will be a growing worldwide market uncertainty with disruptive changes in technology, such as automated driving. This has to be viewed with the backdrop of the immense product variety in the automotive industry with, e.g., 3, 347,807,348,000,000,000,000,000 options for the Mercedes E-Class (Pil & Holweg, 2004) in mind.

1.2. Problem statement

From a practical perspective, the need for advanced prescriptive analytics approaches is clear. However, research has long ignored the rising opportunities and importance to combine analytics with classical operations research and management science approaches (Mortenson et al., 2015).

This thesis seeks to tackle this research gap by establishing advanced decision support systems for several problems that involve fixed costs in inventory management or production planning. In classical inventory management theory, research predominantly considers two cases: either determine an optimal policy or optimize the parameters of a given policy.

We first consider the investigation of the optimal inventory control policy when fixed costs vary cyclically. In operational inventory management, fixed costs occur when an order for one or several products is placed, regardless of the individual quantities. When looking at a problem setting for one individual product, the (s, S) inventory control rule is optimal under certain assumptions. Based on this rule, a retailer executes an order whenever his inventory drops below the reorder level s. The quantity to be ordered is set in such a way that it fills up the inventory to the order-up-to level S. While this finding is based on publications from the 1950s and 1960s, one crucial impractical assumption is made: fixed costs may not increase over time. However, in practice, there are many applications where this assumption does not hold. For instance, deliveries might be scheduled for fix dates as, e.g., in retail or for the distribution of spare parts. If low inventory levels necessitate placing an order between scheduled deliveries, fixed costs such as truck capacities or warehouse picking routes are not fully exploited and hence economies of scale reduce, which leads to higher costs on the level of an individual product. The optimal policy structure for this cost setting is still unknown.

While the investigation of single-product problem settings is valuable for generating theoretical insights, practical applications mostly deal with multiple products in a company. If an order of products incurs (major) fixed costs for multiple products, the so-called *joint replenishment problem (JRP)* comes up. The key to the problem is to decide when it is optimal to place a combined order for multiple products, although the timing might not be optimal for each product on an individual level. In practice, setting optimal parameters for given (simple) inventory control policies is most relevant. We hence consider two delivery concepts and provide approaches to set their parameters.

One basic concept for inventory control in retail is the implementation of fixed delivery rhythms (e.g., delivering on a Monday, Thursday and Saturday or only Wednesdays), which are allocated to products and retail stores individually. In practice, managers use rule-of-thumb and negotiations between decentral units to arrive at an allocation of these delivery rhythms. However, retail demand is highly uncertain and non-stationary and distributions networks extending over a large number of retail stores and thousands of products are complex in nature. Hence, decision support systems, which need to suit the particularities of the retailer, are needed to set delivery rhythms in an automated way. Other inventory control mechanisms for tackling the joint replenishment problem might be worth considering for the retailer, such as the (s, c, S) or can-order policy. Using this approach, orders are triggered by the reorder-level s as in the (s, S) policy, but might also be triggered by the can-order level c, which is the case when the inventory level drops below c and at least one other product has already been ordered.

Another important consideration in the retail industry is the handling effort. The store handling accounts for 38 % and the warehouse handling for 28 % of the logistics costs in retail (Van Zelst et al., 2009). At the warehouse level, manual effort is needed especially for order picking, while in the retail store several processes require manual handling. Shelf replenishment is the most important in that regard. When goods arrive at the retail store, they are used for filling up the store shelves. If the shelf capacity is not sufficient, the overflow needs to be stored in the backroom and used for later replenishment, which leads to a double handling of goods (*the backroom effect (BRE)*).

To model demand uncertainty, data-driven approaches directly use historical data samples for optimization. Thus, there is no need to find theoretical parametric probability distributions and their parameters need not be estimated. Hence, the input is used as-is, without any assumptions on its parametric distribution and it remains easily interpretable for managers. Models that use the approach might be of mixed integer linear type, which makes the addition of practical constraints easy.

Fixed costs, overload situations and unsteady material consumption in operative production environments depend on the sequence they are produced in. OEMs can decide almost without restrictions how to sequence the items to be produced. Also, they want to streamline inventories and hence, let suppliers deliver *just-in-sequence (JIS)*. Consequently, module suppliers who deliver JIS on mixed-model assembly lines are most likely producing in the same sequence as the OEM, although this might not be optimal for them. To introduce flexibility into the supplier's production setting, resequencing, where the OEM sequence is altered to better fit the supplier, is an option. Yet, any restoration of the original sequence has to be done efficiently, which is due to the time pressure in just-in-sequence settings and the otherwise too high restoration efforts. However, smart resequencing of products during assembly might allow for the transformation of a reactive to a decision-driven assembly system for the supplier.

The methodological contribution of this thesis lies in the mixed integer linear program (MILP)-based optimization, which covers uncertainty yet allows simple linear models and the possibility to add constraints. This is especially important for practical applications, so as to enable a fit of the model to the individual situation of a company. Furthermore, if we use the data-driven approach no distributional assumptions regarding the uncertainty have to be made.

Our managerial contributions provide answers to the following research questions:

- How can operations research methods be used to derive advanced decision support in digitized supply chains?

- Does the consideration of uncertainty in the decision-making have any impact on the operational performance?
- How can innovative operations concepts tackle the requirements for cost-efficiency and practical complexity?
- What is the implication of various forms of fixed order costs (time-dependent, multi-item or sequence-dependent) for decision-making?

1.3. Outline

The remainder of this thesis is structured as follows.

We investigate optimal stochastic inventory policies when fixed costs vary cyclically over time in Chapter 2. Fixed costs in inventory management under uncertain demands lead, under some assumptions, to the optimality of (s, S) inventory control rules. We generalize existing proofs, which assume fixed costs to be non-increasing over time, by allowing costs to change cyclically in one period of an order cycle. We show that the optimal policy is of a complex form with potentially multiple order areas in the low-cost period, but reduces to the classic (s_j, S_j) policy for all other periods (with j as the index of the period). We compare several simple inventory policies with the optimal one in a controlled numerical study to get insights into the applicability of the heuristics. This chapter is based on Taube and Minner (2017b).

The first retail application, introduced in Chapter 3, investigates the allocation of fixed delivery rhythms and order-up-to levels for several products and stores. We consider a three-echelon supply chain with a warehouse, several stores, and shelves at the stores. We include handling efforts at all echelons and investigate data-driven approaches to determine cost-minimal allocations of rhythms and order-up-to levels per product and store. Cyclic ordering allows orders which are placed at the end of the time horizon to serve demands that occur at the beginning. This leads to an optimization of the initial inventories. We implement several hierarchical decomposition approaches as MILPs, a genetic algorithm and a savings heuristic. The genetic algorithm is self-adaptive, i.e., it sets its parameters automatically based on a metacalibration approach. We compare the model variants against several benchmarks in a controlled numerical study and the most promising ones in a case study based on data provided by a European retailer. This chapter is based on Taube and Minner (2017a).

The second retail application, in Chapter 4, considers the usage of an (s, c, S, nq) inventory policy, which uses the traditional (s, c, S) policy, but includes the constraint that only integer multiples n of a fixed case pack size q may be ordered. We present a MILP formulation to determine the optimal parameters, using, as before, a data-driven approach with cyclic ordering. Furthermore, we take the backroom effect into account. In a controlled numerical study, we compare our model against simplified benchmarks to investigate the cost benefits when the backroom effect or the stochasticity in the demand are included, or when the (s, c, S), rather than an (s, S) policy, is applied. This chapter is based on Turgut et al. (2017).

In Chapter 5, we consider (first-tier) suppliers who deliver just-in-sequence to an OEM. The supplier produces on mixed-model assembly lines in the original sequence as-is. We suggest an approach that resequences while ensuring a first-in-first-out restoration by the use of parallel lines after assembly (so-called mix banks). We provide a MILP formulation and a limited lookahead heuristic for modeling this problem setting. Based on an adaptable formulation, we consider three objectives from the sequencing literature: load balancing, material leveling and color batching. We apply our model and the heuristic on two controlled testbeds and compare the results with the production of the OEM sequence as-is. This chapter is based on Taube and Minner (2017c).

The conclusion in Chapter 6 presents a summary and the insights of the thesis and points out the potential for future research.

Chapter 2.

Optimal Inventory Control With Cyclic Fixed Order Costs

There are several applications where fixed order costs change in a cyclic way over time, e.g. with smaller fixed costs in one period of the time horizon than in all other periods. The literature on optimal inventory control policies under fixed order costs does not account for this. We close this gap in inventory theory by generalizing existing proofs for optimal inventory policies and, for the high-cost periods in our setting, extending the optimality of (s, S) policies. We consider a periodic review single-item inventory model under stochastic demand. Every m periods, in the regular order period, fixed costs are K, when ordering. In the periods in-between, the intraperiods, higher fixed order costs of $L > K$ apply. We show that the optimal policy is complex in the regular order period and of an (s_j, S_j) type policy in the intraperiods (with j as an index for the intraperiods). We describe and prove this optimal policy based on the notion of K-convexity and the optimal ordering behavior in the presence of non-K-convex cost functions. Although fixed order costs change cyclically, the cost functions of the intraperiods are K-convex and thus lead to optimal (s, S) policies. The cost function in the regular order period does not fulfill K-convexity, which leads to the more complex order structure with multiple order areas. In a numerical study, we compare the optimal inventory policy with several benchmark policies. We find that a major driver of the cost deviations between the optimal policy and the benchmarks is based on a forward-buying effect that aims at reducing the probability of ordering in the intraperiods while at the same time increasing the probability of ordering in the regular order period. The cost differences between the optimal and a classic period-dependent (s, S) policy are negligible. A four-parameter (s, S, r, R) policy, with a reorder point (s) and an order-up-to level (S) for all intraperiods and different parameters (r and R) for the regular order period shows good performance with cost differences of 0.9 % on average.

2.1. Introduction

In the context of stochastic inventory control, the optimality of (s, S) policies under rather general assumptions is widely known. Although generalizations in many directions (e.g. random discounts, Markovian demands, etc.) have been made for more applications, one crucial assumption is that fixed costs do not increase over time (or at least not in their expectation) (see Iglehart, 1963; Scarf, 1959; Sethi & Cheng, 1997; Veinott & Wagner, 1965). This assumption is, however, violated for several practical problems.

In order to reduce the complexity of constantly changing delivery times in the retail industry, stores or products (groups) are often assigned to a fixed delivery schedule, e.g. one delivery per week. This fixed delivery schedule is set in such a way that it exploits economies of scale for the fixed costs of order handling in the warehouse and the transport costs to all stores (e.g., Gaur & Fisher, 2004). Within a store, however, goods might run out between two scheduled deliveries and have to be reordered. As these orders will not be optimally synchronized with the orders of other stores, the effects of economies of scale vanish and result in higher fixed costs per order. As a generalization of the retail problem above, consider the *joint replenishment problem* where an individual product is optimized under given order schedules (and the associated setup costs) of all other products. Then, a cyclic cost structure is present whenever managers are allowed to place single orders in between those schedules. Another example is the spare parts replenishment process of an original equipment manufacturer to its dealers. To minimize fixed costs, dealers are asked to order in bulk only once per week on a fixed day. However, due to the critical nature of spare parts, small "express" orders are regularly ordered in between. Naturally, those small orders incur larger fixed costs per order as neither picking processes in the warehouse nor transportation tours are highly utilized.

In production, schedules are constructed, among other things, to reduce setup times in order to fulfill a relatively stable customer demand for the end-product. However, due to the highly stochastic nature of demand, e.g., for spare parts, it might be necessary to insert an ad-hoc production of certain parts if not available on stock. As these ad-hoc setups disturb the regular production schedule, indirect costs of influenced production lots planned for the future have to be accounted for on top of the direct setup costs. This again leads to higher fixed order costs for those non-regular setups.

The problem setting of cyclic fixed costs also encompasses the general possibility of *multiple supply modes or suppliers*, where not all supply options are available in all periods. There might, for example, be a cheap cargo train available for transport once a week, while during all other days, a more expensive truck delivery has to be used. The setting includes order systems with deterministic *discount opportunities*, where fixed costs are known to drop and increase in a cyclic manner, e.g., when a vendor offers reduced fixed order costs for end-of-season sales.

By taking the setting above into account and explicitly considering that fixed costs might increase over time, we arrive at a generalization of existing studies regarding the (s, S) policy (e.g., of Sethi & Cheng, 1997). We include cyclical fixed order costs, such that there is one regular order period in each cycle (of, e.g., one week), where low fixed order costs of K are incurred. All other periods (so-called intraperiods) have higher fixed order costs of L. We show that the optimal policy is not necessarily an (s, S) policy, but of a more complex type in the order period with low fixed costs. The policy is an (s_j, S_j) policy in all other periods (with j as an index for the intraperiods).

The remainder of this chapter is structured as follows. We start with an overview of the relevant literature in Section 2.2. In Section 2.3, we state the problem formally and give an illustrative example of the structure of the optimal policy. We describe and prove the optimal order policy in Section 2.4, along with some extensions and with finite and infinite horizon settings. In Section 2.5, we provide a numerical study with insights into the performance of several simpler benchmark policies against the optimal one. Section 2.6 concludes the chapter with a brief summary and some outlook on further research.

2.2. Literature

After Arrow et al. (1951) introduced the stochastic inventory control problem and the (s, S) policy, many publications dealt with proving optimality for cases of similar inventory control settings. The basic inventory problem considers a single-stage and single-item problem where fixed order costs prevail. The seminal paper of Scarf (1959), directly extending Arrow et al. (1951) and introducing K-convexity, as well as its successors like Veinott and Wagner (1965), Song and Zipkin (1993) or Sethi and Cheng (1997) generalize these findings (e.g., for Markovian demands, positive lead times, etc.). For a definition of K-convexity see 2.A. These publications, however, require fixed costs to be either constant or at least non-increasing in their values over time (or in their expected values of state-dependent order fixed costs for the Markovian demand case). As there exist several applications where fixed costs rise and fall in a cyclical pattern, this assumption is violated which reveals an important gap in inventory theory that we close with our investigation.

The above-mentioned publications use K-convexity in their proofs and thus limit their cases to K-convex cost functions. For settings where K-convexity is not given, other notions have been used in the inventory control literature. In the area of capacity expansion/reduction and cash balancing problems, (K_1, K_2)-convexity (Ye & Duenyas, 2007) and *weak*-(K_1, K_2)-convexity (Semple, 2007) have been introduced. These problems have in common that there are two different fixed costs for buying and selling. The optimal policies obtained are to some degree similar to the optimal policy found in our case, in that they show multiple buying and selling areas. However, those notions are not applicable to problems with cyclic fixed order costs as the decision incurring the fixed costs is placed in the same period for both, buying and selling. Also, fixed costs are assumed to be non-increasing over time. *sym*-K-convexity is introduced by X. Chen and Simchi-Levi (2004) to characterize the optimality of an (s, S, p) policy for integrated inventory control and price setting with additive random demand, again assuming non-increasing fixed costs. Porteus (1971) introduced $quasi - K$-convexity for concave increasing ordering costs. This can be applied in a setting of multiple suppliers with different fixed and variable order costs. As opposed to our setting, all supplier options are available in every period, which leads to the same concave order cost structure for all periods. Gallego and Sethi (2005) extend K-convexity to order decisions placed for multiple products. In their approach, joint replenishment effects can be obtained when multiple products are ordered together and thus K-convexity is expanded to the $\mathbb{R}^n$ space.

In the dual sourcing literature stream, Johansen and Thorstenson (2014) include different fixed costs for the regular and emergency order mode and propose an (s, Q) policy for the regular and an (s, S) policy for the emergency order mode. They use a policy iteration algorithm to obtain optimal parameters, yet do not provide an analytical proof for an optimal policy. Chiang (2003) integrates fixed costs as being zero either for both or at least for the regular supply mode. The resulting policy has two order-up-to levels (one for the regular and one for the emergency supply) and a reorder point for both modes. Jain et al. (2011) apply varying fixed costs for both supply modes and characterize the complex optimal order policy. They indicate that this complex policy reduces to an (s, S) policy when the difference in fixed order costs of choosing one order mode over the other is negligible. They numerically show that the employment of a pure (s, S) strategy yields good approximations. The difference to our setting is that both order modes are available in all periods and that it is allowed to split orders between both modes.

Another related research area is the single-item inventory problem with random discount opportunities in a continuous review setting. In this scenario, it is assumed that possibilities with reduced costs for ordering (fixed and/or variable costs) occur randomly. Federgruen et al. (1984) conjecture that combined orders of multiple products occur randomly based on an exponential distribution and thus, for a single item, order costs are also reduced randomly due to economies of scale. Simple, yet not necessarily optimal, policies are of an (s, c, S) type, where s and S are used as reorder point and order-up-to level. The parameter c states a can-order point that is considered when a random discount opportunity is at hand. Hurter and Kaminsky (1968) introduce an inventory problem with random discount opportunities and provide an algorithm to determine the optimal parameters for an (s, c, S) policy. However, they do not prove the optimality of the policy. The optimal policies when variable and fixed order costs are changing for discount orders are of (r, R, d, D) type with reorder points (order-up-to levels) r (R) for the regular and d (D) for the discount orders (see Y. Feng & Sun, 2001). However, all papers in the area of random discount opportunities have in common that demand is (compound) Poisson and the discount might happen in any period, thus they do not have a cyclic cost structure. Zipkin (1989) investigates optimal order policies under periodically changing variable costs, which results in a period-dependent base-stock policy. Our work will extend this direction by focusing on cyclical changing fixed order costs.

2.3. Problem Formulation

We consider a finite horizon with $n = 1, \ldots, N$ discrete order cycles, each covering m periods. Later we will generalize to infinite horizon problems. We assume that orders are placed at the beginning of each period and arrive instantaneously (lead time is zero), followed by i.i.d. random demand ξ with distribution density ϕ. Note that in the following analyses and proofs, we describe the cost functions and Bellman equations that assume continuous distributions, but the same holds for discrete distributions, which we will use for our numerical examples. There are two different fixed order costs. K applies in the regular order period every m periods and fixed order costs L $(K < L)$ occur in the intraperiods $j \in \{1, \ldots, m-1\}$. Let the period-dependent fixed order costs $K_j = L$ for all $j = 1, \ldots, m-1$, $K_m = K$ and, for the convenience of writing, $K_{m+1} = K_1 = L$. Variable costs c are incurred for any item ordered. They do not differ between the two order modes. Positive inventory at the end of the period incurs costs of c_H per unit and period and any demand that cannot be fulfilled is backlogged at a cost c_P per unit and period. $0 < \alpha \leq 1$ is the one-period discount factor.

Let x be the inventory on hand before and y the inventory on hand after ordering.

The expected single period holding and penalty cost function $\mathcal{L}(y)$ is

$$\mathcal{L}(y) = \int_0^y c_H(y-\xi)\phi(\xi)d\xi + \int_y^\infty c_P(\xi - y)\phi(\xi)d\xi. \tag{2.1}$$

When assuming an N cycle problem as described above, we want to find the policy that minimizes the expected costs in cycle n and period j and for an inventory state before ordering x (see, for the case with constant fixed order costs, Porteus, 2002). We define this optimal expected costs as

$$f_{n,j}(x) = \min_{y \geq x} \left\{ c(y-x) + \delta(y-x)K_j + \mathcal{L}(y) + \alpha \int_0^\infty f_{\theta(n,j)}(y-\xi)\phi(\xi)d(\xi) \right\}$$

where $\delta(a) = 1$ if $a > 0$, and 0 otherwise and $\theta(n,j) = \begin{cases} (n, j+1) & \text{for } j \leq m-1 \\ (n+1, 1) & \text{for } j = m \end{cases}$ is a one-period time shift operator that either points to the next period in a cycle or to the first period of the next cycle.

We furthermore rewrite the optimal expected costs $f_{n,j}(x)$ as

$$f_{n,j}(x) = -cx + \min \left\{ G_{n,j}(x), \min_{y > x} \{K_j + G_{n,j}(y)\} \right\} \tag{2.2}$$

where

$$G_{n,j}(y) = cy + \mathcal{L}(y) + \alpha \int_0^\infty f_{\theta(n,j)}(y-\xi)\phi(\xi)d(\xi). \tag{2.3}$$

The optimal decision for inventory state x in cycle n and period j is found by determining the minimum of the value $G_{n,j}(x)$ and the minimizing value of $G_{n,j}(y)$ to the right of x, i.e., $y > x$, plus the fixed order costs. We get the optimal order quantity $u^*(x) = y - x$ in this case. Hence, we can limit our investigations regarding the optimal policy by investigating functions $G_{n,j}$ as in Scarf (1959) and Porteus (2002).

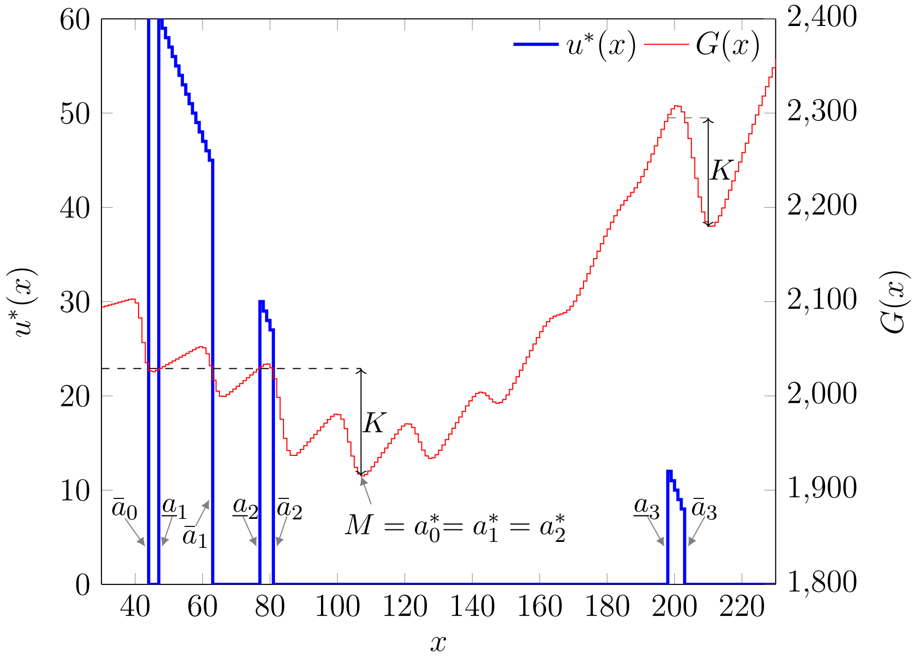
As we consider a finite horizon problem at this point, let f_{TE} be a terminal value function that is assumed to be continuous and convex, and that is charged at the end of period m in cycle N. This terminal value function represents the charge that is incurred for any leftover inventory or backlogged demand at the end of the finite time horizon. Without loss of generality, let us assume, as in Porteus (2002), f_{TE} to be

$$f_{TE}(x) = f_{N,m+1} = -cx$$

i.e. all positive inventory is salvaged with the unit cost and all backlog has to be produced at the unit cost.

To illustrate the structure of the optimal policy in the regular order period m, consider the optimal order quantities $u^*(x)$ in an example with discrete demand shown in Figure 2.1 (see the blue colored stairs plot), with the parameters $c_H = 1$, $c_P = L = 260$, $K = 114$ and a discretized gamma distribution with $\mu = 20, \sigma = 1$ and an infinite number of order cycles ($N = \infty$). To get the discrete probability distribution for the demands ($\phi_D(\cdot)$), we use the continuous cumulative distribution function $\Phi_C(\cdot)$ and set $\phi_D(0) = \phi_C(0)$, $\phi_D(x) = \Phi_C(x) - \Phi_C(x-1)$ for all $x = 1, \ldots, \bar{D} - 1$ and $\phi_C(\bar{D}) = 1 - \sum_{x=0}^{\bar{D}-1} \phi_D(x)$. $\bar{D}$ is the upper bound on the demand, set to the 99th percentile of the continuous cumulative distribution function rounded up to the next integer. The red colored stairs plot represents the cost function $G_{n,m}(x)$ in the regular order period. We observe that for net inventory $x \leq 43$, $47 \leq x \leq 62$ and $77 \leq x \leq 80$, it is optimal to order up to $y = 107$. However, with a net inventory level of $44 \leq x \leq 46$ and $63 \leq x \leq 76$, it is optimal to stay put and order nothing. For the three order areas with $x \leq 80$, there exists one order-up-to level that is the same for the three order areas. Each order area is defined by a reorder and a stay-put point. A reorder point is the inventory level below which an order is placed as long as the inventory level does not drop below the stay-put point. Surprisingly, on the far right of the figure, we can see another order area, where $198 \leq x \leq 202$ and it is optimal to order up to $y = 210$, i.e. there exists another order area with a second order-up-to level. This is due to the sharp drop in the cost function between inventory level 198 and 210 of more than K units, which is why it is more cost-attractive to order with fixed costs of K than to stay put.

Figure 2.1.: G-function and optimal order decisions u^* in the regular order period



The existence of the order areas can be explained by the following intuition. The regular order period is cost-attractive in terms of fixed order costs. Now consider that we are in inventory state x_1. At x_1, we decide to order additional quantities so as to minimize the probability that an order in a future intraperiod has to be placed. I.e., we accept higher expected holding costs to realize expected savings in fixed costs. We refer to this as the *forward-buying* effect. This decision might be optimal up to a certain inventory on-hand $x_2 > x_1$ where it is not optimal to order any more. At x_2, it would not be optimal to order anything for the regular order period and hence the additional costs of ordering to cover the demand of future intraperiods (i.e., forward-buying) plus additional expected holding costs outweigh the expected savings in fixed order costs. When this happens, the reorder point of the order area is reached. If we further assume a larger inventory on hand $x_3 > x_2$, we have higher expected holding costs than before, which occur no matter if we order in the regular order period or not. Hence, any decision to order is again justified if the expected savings in fixed costs are higher than the additional expected holdings costs, which come on top of what the current inventory state x_3 would incur anyway. At that point, we reach another order area and its stay-put level.

Any of the inventory states across all order areas in the example might be reached in a finite-horizon problem, depending on the starting inventory in the first period. However, for the steady-state in the infinite horizon setting, the inventory states in the order area around $y = 198$ will become transient, as inventory levels after ordering will never exceed $y = 107$ after an order from any other order area has been triggered.

In the next section, we will prove that the optimal policy for the intra-periods does not follow the complex order pattern we described above for the regular order period. Instead it shows the traditional period-dependent (s_j, S_j) policy structure.

2.4. Optimal Policy

To characterize the stay-put, reorder and order-up-to level for each order area, we define critical points along a continuous function $g(x)$.

Definition 2.1 (CRITICAL POINTS). *For a continuous function $g(x)$, define*

(i) $M \in \inf_{x \in [-\infty, \infty]} g(x)$

(ii) $\underline{a}_i := \inf \left\{ \bar{a}_{i-1} < x \leq \infty : g(x) \geq g(y) + K, y : \inf_{y \in [x, \infty]} g(x) \right\}$

(iii) $a_i^* := \inf_{x \in [\underline{a}_i, \infty]} g(x)$

(iv) $\bar{a}_i := \sup \left\{ \underline{a}_i \leq x \leq a_i^* : g(y) \geq g(a_i^*) + K \text{ for all } \underline{a}_i \leq y \leq x \right\}$

(v) $\underline{a}_0 := -\infty$

(vi) $\underline{A} = \inf_{i \in I} \{\bar{a}_i\}$

(vii) $\bar{A} = \sup_{i \in I}\{\bar{a}_i\}$

For a visual representation of these points, see Figure 2.1. Consider M as the global minimizer of $g(x)$, $i \in I$ as the set of all relevant order areas, $\underline{a}_i, \bar{a}_i, a_i^*$ as the ith stay-put, reorder and order-up-to level of order area i, $\underline{A}$ as the level below which it is always optimal to order and $\bar{A}$ as the level above which it is never optimal to order. Note that $a_i^* = M$ for all $i \in I | \bar{a}_i \leq M$, i.e. the global minimum M represents the order-up-to level for all order areas to the left of M. We can see that $\underline{a}_i$ and $\bar{a}_i$ form the different order areas, where it is optimal to order. $\bar{a}_i$ is the first point left of the next stay-put level $\underline{a}_{i+1}$, where $G_{n,j}$ increases by more than K with respect to the minimum level right of $\bar{a}_i$, therefore indicating a reorder level. Going further to the left $\underline{a}_i$ is the last point before this cost difference falls below K, thus indicating the stay-put level. This is different to the traditional (s, S) order policies. The reason lies in the values of $G_{n,m}(x)$, which fall below $G_{n,m}(M) + K$ to the left of M and thus make ordering unattractive in some areas.

To characterize the optimal policy, we first establish an important proposition, since we need to show that, although $G_{n,m}$ is not K-convex, the function is still L-convex.

Proposition 2.1. *Let L and K be non-negative, with $L \geq K$ and $g : R \to R$ be an L-convex continuous function such that $g(x) \to +\infty$ as $x \to +\infty$. Let the notation $g(-\infty)$ denote the extended real number $\liminf_{x \to -\infty} g(x)$ and $\delta(x) = 1$ if $x > 0$ and 0 otherwise. Let*

$$g^* = \inf_{x \in (-\infty, \infty)} g(x) > -\infty \tag{2.4}$$

Then

(i) $h(x) \equiv \inf_{y \geq x, -\infty < y < \infty} [K\delta(y - x) + g(y)] = \left\{ \begin{array}{ll} K + g(M) & \text{for } x < \underline{A} \\ K + g(a_i^*) & \text{for } i \in I, x \in (\underline{a}_i, \bar{a}_i) \\ g(x) & \text{for } i \in I, x \in [\bar{a}_{i-1}, \underline{a}_i] \\ g(x) & \text{for } \bar{A} \leq x \leq \infty \end{array} \right\}$

and $h : R \to R$ is continuous;

(ii) *h is L-convex on $(-\infty, \infty)$*

(iii) $k(x) \equiv \inf_{y \geq x, -\infty < y < \infty} [L\delta(y - x) + g(y)] = \begin{cases} L + g(S) & \text{for } x < s \\ g(x) & \text{for } s \leq x < \infty \end{cases}$ *and $k : R \to R$ is continuous;*

(iv) k is L-convex on $(-\infty, \infty)$

Proof. See Appendix 2.B.1. □

In our problem setting, $g(x)$ can be substituted by $G_{n,j}(x)$. $h(x)$ and $k(x)$ can be substituted by the Bellman equations of (2.2) for the regular order period and the intraperiods. When we apply Proposition 2.1 to these substituted cost functions, (i) introduces the optimal order decisions in the regular order period. (ii) proves that L-convexity is preserved under this policy. The proof of (ii) establishes that the definition of K-convexity from Sethi and Cheng (1997) holds for any relevant combination of points on function $h(x)$ as defined in (i).

Based on the preserved L-convexity, we can now use the standard arguments as in Scarf (1959) and Sethi and Cheng (1997) to show that an (s, S) policy is optimal in the intraperiods. We define the optimal inventory control policy for the intraperiods and for the regular order period in the following theorem.

Theorem 2.1. *(i) In the regular order period m of order cycle n, there exist critical points such that, for a given inventory state x, the optimal inventory after ordering is*

$$y_{n,m} = \begin{cases} M_n & \text{for } x \in (-\infty, \underline{A}_n) \\ a^*_{n,i} & \text{for } i \in I_n, x \in (\underline{a}_{n,i}, \bar{a}_{n,i}) \\ x & \text{for } i \in I_n, x \in [\bar{a}_{n,i-1}, \underline{a}_{n,i}] \\ x & \text{for } x \in [\bar{A}_n, \infty] \end{cases}$$

(ii) For all intraperiods $j = 1, \ldots, m-1$ of order cycle n, there exists an optimal $(s_{n,j}, S_{n,j})$ policy with critical numbers $s_{n,j}$ and $S_{n,j}$, such that, for a given inventory state x, the optimal inventory after ordering is

$$y_{n,j} = \begin{cases} S_{n,j} & \text{for } x \in [-\infty, s_{n,j}) \\ x & \text{for } x \in [s_{n,j}, \infty] \end{cases}$$

Proof. See Appendix 2.B.2. □

The proof for the intraperiod policy uses Proposition 2.1 and the basic properties of K-convexity of Scarf (1959) and Sethi and Cheng (1997). It proves by induction that L-convexity is given in every cycle and intraperiod and thus an (s, S) policy is optimal.

2.4.1. Parameter Bounds

For the optimal policy, bounds on the optimal parameters like in Veinott (1966) hold, with some exceptions as explained below.

Proposition 2.2. *For each period $j = 1, \dots, m$, let numbers $\underline{s}_j$, $\underline{S}_j$ and $\bar{S}_j$ be defined to fulfill:*

$$\underline{S}_j = \inf_{y \in [-\infty, \infty]} \mathcal{L}(y)$$

$$\mathcal{L}(\bar{S}_j) = \mathcal{L}(\underline{S}_j) + K_{j+1}$$

$$\mathcal{L}(\underline{s}_j) = \mathcal{L}(\underline{S}_j) + K_j$$

and additionally for each period $j = 1, \dots, m-1$ let number $\bar{s}_j$ be defined to fulfill

$$\mathcal{L}(\bar{s}_j) = \mathcal{L}(\underline{S}_j) + (K_j - K_{j+1}).$$

Then,

1. *for the regular order period m the optimal parameters of $\underline{A_n}$, $\underline{a}_{n,i}$, $\bar{a}_{n,i}$, $a^*_{n,i}$ and $\bar{A}_n$ are bounded by parameters*

$$\underline{s}_m \leq \underline{A}_n \leq \underline{a}_{n,i} \leq \bar{a}_{n,i} \leq a^*_{n,i} \leq \bar{A}_n \leq \bar{S}_m$$

   and

2. *for the $(s_{n,j}, S_{n,j})$ policy in the intraperiods, the optimal parameters $s_{n,j}$ and $S_{n,j}$ are bounded by parameters*

$$\underline{s}_j \leq s_{n,j} \leq \bar{s}_j \leq \underline{S}_j \leq S_{n,j} \leq \bar{S}_j.$$

Proof. The assumptions (i)-(v) of Veinott (1966) are valid in all periods and due to the assumption on the left-over stock after demand and the time-independent definition of the one-period expected holding and shortage costs $\mathcal{L}(y)$. For the regular order period m, these assumptions suffice to establish $\underline{s}_m$ and $\bar{S}_m$ as bounds. Note that assumption (vi) of Veinott (1966) imposes that $K_j \geq K_{j+1}$, which is true for $j = 1, \dots, m-1$, but not for $j = m$. This is why $\bar{s}_j$ and $\underline{S}_j$ can be constructed as bounds for the intraperiods but not for the regular order period. For the rest of the proof see Veinott (1966). □

2.4.2. Infinite Horizon

In this subsection, we consider the infinite horizon setting, with $N \to \infty$. As before, we make decisions on the inventory after ordering in order cycle n and period j and let $\mathbb{y} = (y_{n,j}, y_{n,j+1}, \ldots, y_{n+1,1}, \ldots)$ be the vector of those future decisions. We want to minimize the following cost function (see Sethi & Cheng, 1997):

$$\begin{aligned} J_{n,j}(x, \mathbb{y}) = &\sum_{k=j}^{m} \alpha^{k-j} E\left[c(y_{n,k} - x_{n,k}) + \delta(y_{n,k} - x_{n,k})K_k + \mathcal{L}(y_{n,k})\right] + \\ &\sum_{k=1}^{m} \sum_{l=n+1}^{\infty} \alpha^{(l-n)m-j+k} E\left[c(y_{l,k} - x_{l,k}) + \delta(y_{l,k} - x_{l,k})K_k + \mathcal{L}(y_{l,k})\right] \end{aligned} \tag{2.5}$$

(2.5) contains the expected discounted costs of all future decisions on the inventory after ordering $\mathbb{y}$ at a given order cycle n, period j and inventory on hand x. The first term considers all periods until the end of the current order cycle n and the second term all future order cycles.

We use the l order cycle truncation of the infinite horizon problem in (2.5) in the next steps, as in Sethi and Cheng (1997). The truncation represents a finite approximation of the infinite horizon. Let $J_{n,j,l}$ be the l order cycle truncation of (2.5), hence

$$\begin{aligned} J_{n,j,l}(x, \mathbb{y}) = &\sum_{k=j}^{m} \alpha^{k-j} E\left[c(y_{n,k} - x_{n,k}) + \delta(y_{n,k} - x_{n,k})K_k + \mathcal{L}(y_{n,k})\right] + \\ &\sum_{k=1}^{m} \sum_{q=n+1}^{n+l-1} \alpha^{(q-n)m-j+k} E\left[c(y_{q,k} - x_{q,k}) + \delta(y_{q,k} - x_{q,k})K_k + \mathcal{L}(y_{q,k})\right]. \end{aligned} \tag{2.6}$$

The truncation considers the following periods until the end of order cycle n and the $l-1$ order cycles after n. We define the dynamic programming equations for the minimization of (2.6) as follows:

$$f_{n,j,l}(x) = -cx + \min\left\{G_{n,j,l}(x), \min_{y>x}\left\{K_j + G_{n,j,l}(y)\right\}\right\}$$

with

$$G_{n,j,l}(y) = cy + \mathcal{L}(y) + \alpha \int_0^{\infty} f_{\theta(n,j,l)}(y - \xi)\phi(\xi)d(\xi).$$

$\theta(n,j,l) = \begin{cases} (n, j+1, l) & \text{for } j \leq m-1 \\ (n+1, 1, l-1) & \text{for } j = m \end{cases}$ is again the one-period time shift operator that either points to the next period in a cycle or to the first period of the next cycle, which then reduces the remaining time horizon of the truncation by one order-cycle, i.e., we have $l-1$ instead of l remaining order cycles. Let $f_{n+l,m+1}(x) = f_{TE}$ again be the terminal value at the end of the truncated time horizon.

Using this truncation, we can now prove the following corollary on the optimal inventory control policy.

Corollary 2.1. *(i) In the regular order period m, there exist critical points such that, for a given inventory state x, the optimal inventory after ordering is*

$$y_m = \begin{cases} M & \textit{for } x \in (-\infty, \underline{A}) \\ a_i^* & \textit{for } i \in I, x \in (\underline{a}_i, \bar{a}_i) \\ x & \textit{for } i \in I, x \in [\bar{a}_{i-1}, \underline{a}_i] \\ x & \textit{for } x \in [\bar{A}, \infty] \end{cases} .$$

(ii) For all intraperiods $j = 1, \ldots, m-1$, there exists an optimal (s_j, S_j) policy with critical numbers s_j and S_j, such that, for a given inventory state x, the optimal inventory after ordering is

$$y_j = \begin{cases} S_j & \textit{for } x \in [-\infty, s_j) \\ x & \textit{for } x \in [s_j, \infty] \end{cases} .$$

Proof. We prove that $G_{n,j}$ is L-convex in the infinite horizon setting. Using the same inductive argument as in Appendix 2.B.2, it is obvious that $G_{n,j,l}$ in the finite truncation of the infinite horizon problem is L-convex for all $j = 1, \ldots, m$. Due to the definition of K-convexity and Proposition 2.3.(iii) in Appendix 2.A, the L-convexity of $G_{n,j,l}$ is preserved when we take the limit as $l \to \infty$, assuming that this limit exists. Thus, $G_{n,j}$ of the infinite horizon setting is also L-convex. We next need to show that the limit of $G_{n,j,l}$ exists. We use Theorem 6.1 of Sethi and Cheng (1997). Both the necessary assumptions of variable plus fixed order costs and a convex and asymptotically linear surplus cost function ($\mathcal{L}(y)$) are valid for our setting. Sethi and Cheng (1997) show that

$f_{n,j,l}$ is bounded and approaches $f_{n,j}$ for the infinite horizon problem when $l \to \infty$. Based on the definition of $G_{n,j}$ for the infinite horizon and $G_{n,j,l}$ for the truncated problem, also $G_{n,j,l}$ has a limit and approaches $G_{n,j}$. An optimal order policy thus finds the minimum for (2.5). As $G_{n,m,l}$ approaches $G_{n,m}$, we can apply Proposition 2.1.(i) to obtain the optimal policy for the regular order period m, as described in Corollary 2.1.(i). $G_{n,j}$ is L-convex and we can hence apply Proposition 2.1.(iii) to get the optimal policy for the intraperiods $j = 1, \dots, m-1$, as described in Corollary 2.1.(ii). □

2.4.3. Markov Modulated Demand and Fixed Costs

The result extends to Markov modulated demand as in Sethi and Cheng (1997). We introduce subscript z as a discrete state-of-the-world variable with the total number of states being Z. Let p_{zi} be the transition probability of reaching state i in the next period when being in state z in the current period. We assume that the intraperiod fixed costs fulfill $K_{j,z} \leq \bar{K}_{j+1,z} = \sum_{i=1}^{Z} p_{zi} K_{j+1,i}$ for all $j = 1, \dots, m-2$. I.e. in any state z during the intraperiods, the expected fixed costs over all states that are potentially reached in the next intraperiod are smaller than or equal to those of the current state and period. Note that this assumption includes fixed order costs, which are not state-dependent, yet remain equal or decrease from one intraperiod to the next, i.e. where $K_j \geq K_{j+1}$ for all $j = 1, \dots, m-2$. This extends the constant fixed costs assumption of Section 2.3. To model the cyclical decreasing fixed costs of the regular order period m, we assume that $0 \leq K_{m,z} < \bar{K}_{1,z}$.

Corollary 2.2. *(i) In the regular order period m of order cycle n and state z, there exist critical points such that, for a given inventory state x, the optimal inventory after ordering is*

$$y_{n,m,z} = \begin{cases} M_{n,z} & \text{for } x \in (-\infty, \underline{A}_{n,z}) \\ a^*_{n,i,z} & \text{for } i \in I_{n,z}, x \in (\underline{a}_{n,i,z}, \bar{a}_{n,i,z}) \\ x & \text{for } i \in I_{n,z}, x \in [\bar{a}_{n,i-1,z}, \underline{a}_{n,i,z}] \\ x & \text{for } x \in [\bar{A}_{n,z}, \infty] \end{cases}$$

(ii) For all intraperiods $j = 1, \dots, m-1$ of order cycle n and state z, there exists an optimal $(s_{n,j,z}, S_{n,j,z})$ policy with critical numbers $s_{n,j,z}$ and $S_{n,j,z}$, such that, for a

given inventory state x, the optimal inventory after ordering is

$$y_{n,j,z} = \begin{cases} S_{n,j,z} & \text{for } x \in [-\infty, s_{n,j,z}) \\ x & \text{for } x \in [s_{n,j,z}, \infty] \end{cases}$$

Proof. The corollary is a straightforward extension from Theorem 2.1 and the proof thereof in Appendix 2.B.2. □

Other extensions such as a constant lead time for all periods, no-order periods (i.e., where in some periods of the cycle no orders are allowed), storage and service constraints for the finite and a cyclical demand model for the infinite horizon case with non-increasing fixed order costs can be found in Sethi and Cheng (1997). The extensions are also straightforward applicable in our cost setting.

2.5. Numerical Study

2.5.1. Setup

In this section, we investigate the optimal order policies over a broad parameter set and quantify the performance of several simpler benchmark policies in comparison to the optimal one. This will give us insights into the occurrence and importance of non-(s, S) order policies in the regular order period. At the same time, we will answer the question: How do sub-optimal policies, which might be easier to implement in practice, perform?

We assume that the uncertain demands are gamma distributed because we want to cover a wide range of the coefficient of variation and to avoid negative demand values. A finite number of states reduces complexity for the solution algorithms. Hence, we assume discrete demand with a maximum demand value $\bar{D}$ set to the 99th percentile of the continuous cumulative distribution ($\Phi_C(\cdot)$), rounded to next larger integer. To get a discrete probability distribution function for the demands ($\phi_D(\cdot)$), we use the continuous cumulative distribution function $\Phi_C(\cdot)$ and set $\phi_D(0) = \phi_C(0)$, $\phi_D(x) = \Phi_C(x) - \Phi_C(x-1)$ for all $x = 1, \ldots, \bar{D} - 1$ and $\phi_C(\bar{D}) = 1 - \sum_{x=0}^{\bar{D}-1} \phi_D(x)$. We introduce the finite inventory state space $X = \{\underline{X}, \ldots, \bar{X}\}$ and set $\underline{X} = -m\bar{D}$ and $\bar{X} = m\bar{D}$.

We vary parameters in a full factorial design as follows. We set $\mu \in \{5, 15\}$, the *coefficient of variation (CV)* $\in \{0.1, 0.3, 0.5\}$, the order cycle length $m \in \{2, 3, 5, 7\}$, the *time between orders (TBO)* $\in \{0.5m, 1m, 1.5m, 2m, 5m\}$ and the holding costs $c_H \in \{0.1, 0.5, 1, 2\}$ with $c_H + c_P = 10$ (similiar as in Jain et al., 2011). The TBO is defined as the optimal time interval for ordering, when the assumptions of the classical economic order quantity model apply. We set the high fixed order costs of the intraperiods based on the TBO value $L = \frac{1}{2}(TBO^2 \cdot \mu \cdot c_H)$ and the lower fixed order costs for the regular order period $K \in \{0.1L, 0.2L, \ldots, 0.9L\}$. We assume $\alpha = 1$ and hence optimize under the average cost criterion.

For a finite horizon problem of 3 and 10 order cycles, we compare the optimal policy with the best (s_j, S_j) policy. The (s_j, S_j) policy uses period-dependent parameters for s and S, also in the regular order period, where the (s, S) policy might not be optimal, as shown in Section 2.4. For all finite horizon policies, we consider the resulting average costs over all possible initial inventory states, weighted equally rather than picking only one arbitrary state. Additionally, we consider the infinite horizon problem, where we compare the optimal policy with the best (s_j, S_j), (s, S, r, R), (s, S) and $(Opt|m)$ policy. The (s, S, r, R) policy is limited to the usage of the same reorder (s) and order-up-to level (S) for the intraperiods but has different ones (r and R) for the regular order period. The (s, S) policy, on the other hand, does not distinguish between an intra- and a regular order period and always uses the same parameters. The $(Opt|m)$ policy always only orders in the regular order period, i.e., every m periods.

2.5.2. Solution Algorithm

For the finite horizon problems, we solve the Bellman equations recursively and find values for all required policy parameters within a multi-dimensional search for each period in the (s_j, S_j) policy. For the infinite horizon problems, we use policy iteration algorithms (see, e.g., Tijms, 2003) to obtain the best parameters. We alter the standard policy iteration for the heuristic approaches (i.e., the (s_j, S_j), (s, S, r, R), (s, S) and $(Opt|m)$ policies). For the $(Opt|m)$ policy, we limit the action space in the intraperiods, so that no order is allowed. For the (s_j, S_j), (s, S, r, R) and (s, S) policies, we conduct a two dimensional search for the best improvement step of each reorder and order-up-to level parameter. It is important to consider that the algorithms might not converge in all the cases where simultaneous changes in the parameters of multiple periods lead to worse solutions in the cost determination step. Convergence cannot be guaranteed as the improvement step will not necessarily lead to less or equal costs in the value function for each state. This is because the improvement step searches for the best values of s and S, which optimizes the sum of the value functions of all states. A similar problem has been identified by Johansen and Melchiors (2003) for a modified policy iteration algorithm that aims at finding optimal parameters for an (s, c, S) policy. As in Johansen and Melchiors (2003), we terminate the algorithms when cycles in the resulting order policies occur or policies did not change after the improvement step.

Let k be the long-term average costs of the system that are to be minimized and $V(x, j)$ the relative values of inventory state x in period j. We furthermore define the direct costs as

$$C(x, j, a) = \delta(a)K_j + c(a) + \mathcal{L}(x + a),$$

where a represents the order quantity. We want to find the cost-optimal policy P^*.

2.5.2.1. Optimal Policy and $(Opt|m)$ Policy

The action space for the optimal policy is dependent on the inventory before ordering x and is restricted to $U(x) = \{0, \dots, \bar{X} - x\}$. The policy iteration algorithm of the optimal policy is defined in Algorithm 2.1.

Algorithm 2.1 Policy iteration for the optimal policy

Step 0: Initialization

Determine a first feasible policy P by minimizing the direct costs.
for $x \in X, j = 1, \dots, m$ **do**
  $P(x, j) = \arg \inf_{u \in U(x)} \{C(x, j, u)\}$
end for
Set $\hat{P} := P$ and $k^* := \infty$.

Step 1: Cost determination

Solve the linear equation system for policy P with the unknowns V and k:

$$V(x, j) = C(x, j, P(x, j)) - k + \sum_{\xi=0}^{\infty} V(x + P(x, j) - \xi, \Theta(j))\phi_D(\xi) \qquad x \in X, j \in \{1, \dots, m\}$$

$$V(a, 1) = 0$$

where $\Theta(j) = \begin{cases} j + 1 & \text{for } j \leq m - 1 \\ 1 & \text{for } j = m \end{cases}$ and $a \in X$ is an arbitrary inventory state.
if $k < k^*$ **then**
  $k^* := k$, $P^* := P$
end if

Algorithm 2.1 Policy iteration for the optimal policy (continued)

Step 2: Policy improvement

for $x \in X$, $j = 1, \dots, m$ **do**

$$P(x,j) = \arg \inf_{u \in U(x)} \left\{ C(x,j,u) - k + \sum_{\xi=0}^{\infty} V(x+u-\xi, \Theta(j))\phi_D(\xi) \right\}$$

end for

Step 3: Termination

if $P = \hat{P}$ **then** Terminate.
else
  $\hat{P} := P$ and **go to** Step 1
end if

For the $(Opt|m)$ policy, we proceed similar to Algorithm 2.1, but define the action space period-dependent as $U(x,j) = \{0\}$ for all $j = 1, \dots, m-1$ and $U(x,m) = \{0, \dots, \bar{X} - x\}$.

2.5.2.2. (s_j, S_j) Policy

For the (s_j, S_j) policy, we want to get the optimal parameters s_j^* and S_j^*. We define action $u(x,s,S) = \delta(s-x) \cdot (S-x)$. Furthermore, we introduce set $\mathcal{P}$ as the set of order policies to be able to control for cycles in the generated order policies. The policy-iteration algorithm for the (s_j, S_j) policy is defined in Algorithm 2.2.

Algorithm 2.2 Policy iteration for the (s_j, S_j) policy

Step 0: Initialization

for $j = 1, \dots, m$ **do**

$$(\hat{S}_j, \hat{s}_j) = \arg \inf_{S_j \in X, s_j \in [\underline{X}, S_j]} \left\{ \sum_{x \in X} C(x, j, u(x, s_j, S_j)) \right\}$$

  for $x \in X$ **do**
    $P(x,j) = u(x, \hat{s}_j, \hat{S}_j)$
  end for
end for
Set $k^* := \infty$, $\hat{P} := P$ and $\mathcal{P} := \{P\}$.

Step 1: Cost determination

As in Step 1 of Algorithm 2.1.

Algorithm 2.2 Policy iteration for the (s_j, S_j) policy (continued)

Step 2: Policy improvement

for $j = 1, \ldots, m$ **do**

$(\hat{S}_j, \hat{s}_j) = \arg \inf_{S_j \in X, s_j \in [\underline{X}, S_j]} \left\{ \sum_{x \in X} C(x, j, u(x, s_j, S_j)) - k + \sum_{\xi=0}^{\infty} V(x + u(x, s_j, S_j) - \xi, \Theta(j))\phi_D(\xi) \right\}$

for $x \in X$ **do**

$P(x, j) = u(x, \hat{s}_j, \hat{S}_j)$

end for

end for

Step 3: Termination

if $P \in \mathcal{P}$ **then**

for $j = 1, \ldots, m$ **do**

$s^*_j = \inf\{x \in X : P^*(x, j) = 0\}$, $S^*_j = \sup\{x + P^*(x, j) : x \in X, P^*(x, j) > 0\}$

end for

Terminate.

else

$\hat{P} := P$, $\mathcal{P} := \mathcal{P} + \{P\}$ and **go to** Step 1

end if

2.5.2.3. (s, S, r, R) Policy

For the (s, S, r, R) policy, we want to get the optimal parameters s^*, S^*, r^* and R^*. We define $u(x, s, S)$ and $\mathcal{P}$ as before. The policy-iteration algorithm for the (s, S, r, R) policy is defined in Algorithm 2.3.

Algorithm 2.3 Policy iteration for the (s, S, r, R) policy

Step 0: Initialization

$(\hat{S}, \hat{s}) = \arg \inf_{S \in X, s \in [\underline{X}, S]} \left\{ \sum_{j=1,\ldots,m-1} \sum_{x \in X} C(x, j, u(x, s, S)) \right\}$

for $x \in X$, $j = 1, \ldots, m-1$ **do**

$P(x, j) = u(x, \hat{s}, \hat{S})$

end for

$(\hat{R}, \hat{r}) = \arg \inf_{R \in X, r \in [\underline{X}, R]} \left\{ \sum_{x \in X} C(x, m, u(x, r, R)) \right\}$

for $x \in X$ **do**

$P(x, m) = u(x, \hat{r}, \hat{R})$

end for

Set $k^* := \infty$, $\hat{P} := P$ and $\mathcal{P} := \{P\}$.

Step 1: Cost determination

As in Step 1 of Algorithm 2.1.

Algorithm 2.3 Policy iteration for the (s, S, r, R) policy (continued)

Step 2: Policy improvement

$$(\hat{S}, \hat{s}) = \arg \inf_{S \in X, s \in [\underline{X}, S]} \left\{ \sum_{j=1,\ldots,m-1} \sum_{x \in X} C(x, j, u(x, s, S)) - k + \sum_{\xi=0}^{\infty} V(x + u(x, s, S) - \xi, \Theta(j)) \phi_D(\xi) \right\}$$

for $j = 1, \ldots, m-1$, $x \in X$ **do**

$P(x, j) = u(x, \hat{s}, \hat{S})$

end for

$$(\hat{R}, \hat{r}) = \arg \inf_{R \in X, r \in [\underline{X}, R]} \left\{ \sum_{x \in X} C(x, m, u(x, r, R)) - k + \sum_{\xi=0}^{\infty} V(x + u(x, r, R) - \xi, \Theta(j)) \phi_D(\xi) \right\}$$

for $x \in X$ **do**

$P(x, m) = u(x, \hat{r}, \hat{R})$

end for

Step 3: Termination

if $P \in \mathcal{P}$ **then**

Set $s^* = \inf\{x \in X : P^*(x, 1) = 0\}$, $S^* = \sup\{x + P^*(x, 1) : x \in X, P^*(x, 1) > 0\}$,
$r^* = \inf\{x \in X : P^*(x, m) = 0\}$, $R^* = \sup\{x + P^*(x, m) : x \in X, P^*(x, m) > 0\}$ and terminate.

else

$\hat{P} := P$, $\mathcal{P} := \mathcal{P} + \{P\}$ and **go to** Step 1

end if

2.5.2.4. (s, S) Policy

For the (s, S) policy, we want to get the optimal parameters s^* and S^*. The policy-iteration algorithm for the (s, S) policy is defined in Algorithm 2.4. We define $u(x, s, S)$ and $\mathcal{P}$ as before.

Algorithm 2.4 Policy iteration for the (s, S) policy

Step 0: Initialization

$$(\hat{S}, \hat{s}) = \arg \inf_{S \in X, s \in [\underline{X}, S]} \left\{ \sum_{j=1}^{m} \sum_{x \in X} C(x, j, u(x, s, S)) \right\}$$

for $x \in X$, $j = 1, \ldots, m$ **do**

$P(x, j) = u(x, \hat{s}, \hat{S})$

end for

Set $k^* := \infty$, $\hat{P} := P$ and $\mathcal{P} := \{P\}$.

Step 1: Cost determination

As in Step 1 of Algorithm 2.1.

Step 2: Policy improvement

$$(\hat{S}, \hat{s}) = \arg \inf_{S \in X, s \in [\underline{X}, S]} \left\{ \sum_{j=1}^{m} \sum_{x \in X} C(x, j, u(x, s, S)) - k + \sum_{\xi=0}^{\infty} V(x + u(x, s, S) - \xi, \Theta(j)) \phi_D(\xi) \right\}$$

for $j = 1, \ldots, m$, $x \in X$ **do**

$P(x, j) = u(x, \hat{s}, \hat{S})$

end for

Algorithm 2.4 Policy iteration for the (s, S) policy (continued)

Step 3: Termination

if $P \in \mathcal{P}$ **then**
  Set $s^* = \inf\{x \in X : P^*(x,1) = 0\}$, $S^* = \sup\{x + P^*(x,1) : x \in X, P^*(x,1) > 0\}$ and terminate.
else
  $\hat{P} := P$, $\mathcal{P} := \mathcal{P} + \{P\}$ and **go to** Step 1
end if

2.5.3. Results

2.5.3.1. Probabilities of Ordering

In Table 2.1, we show the effect of the parameter variations on the probability of ordering in either the regular or an intraperiod for the optimal policy and an infinite horizon. The probabilities represent the sum of the steady-state probabilities of all inventory states in which the optimal policy orders a positive amount. I.e., they represent the percentage of periods in which either in the regular or in the intraperiod an order takes place.

We see that the effect a difference in μ has on the order probabilities is negligible.

A more pronounced effect can be observed when we compare the different values for the CV. The higher the CV, the lower the probability of ordering in the regular order period and the higher the probability of ordering in the intraperiod. For a higher CV, higher safety stocks, too, have to be in place. When an order in the regular order period covers the demand of a number of future intraperiods (i.e., through forward-buying), the order quantity and hence the holding costs will increase. This increase in the holding costs makes it less attractive to order in the regular order period, as the fixed costs benefit decreases.

When analyzing the probability of ordering at all (i.e. the sum of the two probabilities), another effect of the CV can be observed. With an increase in the CV, the probability of ordering at all decreases. This is surprising at first sight, as one would expect that higher uncertainty will necessitate having to order more often. However, with an increase in the CV, the probability of larger demands increases. This leads to a decrease in the expected holding costs. The trade-off between expected holding and fixed order costs becomes more favorable for holding inventories and thus the optimal parameters will lead to a decrease in the order probability.

With a rising value for m or for the TBO, the probability for both regular and

Table 2.1.: Average probabilities of ordering (in %)

Parameter	Regular period	Intraperiod	Sum	Parameter	Regular period	Intraperiod	Sum
μ				K/L			
5	8.57	2.79	11.36	0.1	9.94	2.36	12.29
15	8.61	2.70	11.30	0.2	9.71	2.36	12.07
				0.3	9.62	2.37	11.99
CV				0.4	9.41	2.39	11.80
0.1	9.62	2.47	12.09	0.5	8.96	2.46	11.42
0.3	8.58	2.82	11.40	0.6	8.53	2.60	11.12
0.5	7.56	2.94	10.50	0.7	7.85	2.93	10.78
				0.8	7.05	3.35	10.39
m				0.9	6.23	3.88	10.11
2	20.57	5.66	26.23				
3	9.00	3.61	12.62	c_H (c_P)			
5	3.17	1.13	4.30	0.1 (9.9)	8.74	3.18	11.92
7	1.61	0.57	2.18	0.5 (9.5)	8.76	2.76	11.53
				1 (9)	8.57	2.60	11.18
TBO				2 (8)	8.27	2.43	10.70
$0.5m$	9.91	11.02	20.93				
$1m$	9.87	1.07	10.94				
$1.5m$	9.16	0.73	9.88				
$2m$	8.17	0.63	8.80				
$5m$	5.83	0.27	6.10				

intraperiod ordering decreases. This is as expected, as both m and the TBO value directly influence the level of fixed order costs K and L.

The table illustrates two effects of an increase in the ratio K/L. First of all, the probability of ordering in the regular order period decreases and the probability of ordering in an intraperiod increases, as expected. This is due to the fact that the fixed costs benefit of ordering in the regular order period decreases until it is as beneficial to order in an intraperiod as in the regular order period (when $K = L$). However, we also observe that the probability of ordering at all is also decreasing with larger K/L. An increase of K/L means that K is rising while L remains constant. Hence the average fixed costs over all periods increases, which in turn leads to order policies that order less often.

With a rising value for c_H, order probabilities in the regular and intraperiod decrease.

This impact of c_H on the order probabilities is based on two effects. Smaller holding costs lead to larger order quantities and hence less order probability. On the other hand, the out-of-stock penalty c_P is directly connected to the holding costs. Hence, when c_H decreases, c_P increases. The larger penalty, in turn, increases the probability of ordering, so as to ensure that out-of-stock situations are reduced. While the effects of c_H and c_P oppose each other, the penalty costs have a larger effect on the overall order probabilities in our numerical study.

2.5.3.2. Results of the (s_j, S_j) Policy

We investigate the performance of the (s_j, S_j) policy in Table 2.2. Any deviations to the optimal policy are based on the possible multiple order areas in the regular order period as discussed in Section 2.3. The Table is structured to show the one-dimensional effect of each parameter. I.e., every row represents the averages and maximum values over the subset of instances where the particular parameter value applies.

From our numerical study, we find that only in a rare number of cases actual cost differences occur. Only 3.4 % of the 4,320 instances of the numerical study have more than one order area in the optimal solution for the regular order period, and thus might have a potential cost difference between the two policies.

Overall, average cost differences to the optimal policy are < 0.01 % independent of the number of order cycles. Over all instances, a maximum cost deviation of 0.57 % occurs for an instance with 3 order cycles. The parameters of this instance are $\mu = 15$, $CV = 0.1$, $m = 3$, $TBO = 0.5m$, $K/L = 0.1$ and $c_H = 0.1$. This shows that the cost effect of not having multiple order areas in the regular period, as we have them in the optimal policy, is marginal for a wide range of parameter settings.

When looking at the maximum deviations for 3, 10 and infinite order cycles individually, one finds several similarities. The maximum deviations occur when CV, TBO and K/L attain their smallest value, when $m = 3$ and when μ is either large and c_H very small (3 order cycles) or when μ is small and c_H is large (10 and infinite order cycles). Most of those instance characteristics also drive the share of instances with multiple order areas. Only for c_H, a smaller value clearly induces more instances with multiple order areas and the values of μ have only a very limited influence.

Maximum cost deviations become smaller the longer the problem horizon, i.e., the higher the number of order cycles. The effect can be explained by the probability

Table 2.2.: Order areas and average/maximum cost deviations of (s_j, S_j) policy (in %)

Parameter	Value	Order areas 1	2	3	$N=3$ Avg.	Max.	$N=10$ Avg.	Max.	$N=\infty$ Avg.	Max.
μ	5	96.67	2.96	0.37	0.00	0.55	0.00	0.37	0.00	0.42
	15	96.48	2.78	0.74	0.00	0.57	0.00	0.19	0.00	0.00
CV	0.1	90.21	8.13	1.67	0.01	0.57	0.00	0.37	0.00	0.42
	0.3	99.51	0.49	0.00	0.00	0.00	0.00	0.01	0.00	0.00
	0.5	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m	2	97.22	2.78	0.00	0.00	0.30	0.00	0.09	0.00	0.00
	3	94.72	4.26	1.02	0.01	0.57	0.00	0.37	0.00	0.42
	5	96.94	2.41	0.65	0.00	0.05	0.00	0.07	0.00	0.07
	7	97.41	2.04	0.56	0.00	0.01	0.00	0.04	0.00	0.00
TBO	$0.5m$	89.81	9.14	1.04	0.01	0.57	0.00	0.37	0.00	0.42
	$1m$	94.68	3.82	1.50	0.00	0.30	0.00	0.09	0.00	0.00
	$1.5m$	98.73	1.04	0.23	0.00	0.04	0.00	0.02	0.00	0.00
	$2m$	99.65	0.35	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	$5m$	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
K/L	0.1	93.96	4.38	1.67	0.01	0.57	0.00	0.37	0.00	0.42
	0.2	96.46	3.54	0.00	0.00	0.34	0.00	0.14	0.00	0.00
	0.3	96.67	3.13	0.21	0.00	0.21	0.00	0.08	0.00	0.00
	0.4	95.63	4.38	0.00	0.00	0.08	0.00	0.03	0.00	0.00
	0.5	93.13	4.79	2.08	0.00	0.30	0.00	0.09	0.00	0.00
	0.6	96.67	2.71	0.63	0.00	0.10	0.00	0.04	0.00	0.00
	0.7	99.38	0.63	0.00	0.00	0.02	0.00	0.01	0.00	0.00
	0.8	97.71	1.88	0.42	0.00	0.04	0.00	0.02	0.00	0.00
	0.9	99.58	0.42	0.00	0.00	0.02	0.00	0.00	0.00	0.00
c_H (c_P)	0.1 (9.9)	95.09	3.70	1.20	0.00	0.57	0.00	0.19	0.00	0.07
	0.5 (9.5)	95.93	3.33	0.74	0.00	0.55	0.00	0.24	0.00	0.00
	1 (9)	96.94	2.78	0.28	0.00	0.38	0.00	0.17	0.00	0.00
	2 (8)	98.33	1.67	0.00	0.00	0.26	0.00	0.37	0.00	0.42
Total		96.57	2.87	0.56	0.00	0.57	0.00	0.37	0.00	0.42

distribution of the inventory states. The steady-state probabilities of the infinite horizon case show that the states of the order areas become (almost) transient, as we explained in our base example in Figure 2.1 of Section 2.3. The effect of the length of the time horizon (i.e., the number of order cycles) can be explained by the fact that we average over all initial inventory positions in the finite horizon problems. Hence, the probability of getting into those states that are transient in the long run increases and thus also the cost differences between the (s_j, S_j) and the optimal policy increase with a smaller the number of order cycles.

Changes in the mean demand μ have no effect for a time horizon of 3 order cycles, but maximum cost differences are larger for the smaller μ when the time horizon becomes longer. The reason lies in the size of the inventory states. For larger μ, more inventory states are present and, as costs are averaged over all inventory states, cost differences are likely to be smaller than for small μ.

The values for CV show that significant cost differences only occur for the smallest value of 0.1. This is due to our finding that the probabilities to order in the regular order period, and in any period at all, rise with a smaller CV. Hence, the multiple order areas gain more importance.

An increase in m from two to three first has a positive effect and later a negative effect on the maximum cost differences. This is based on several overlapping effects. First of all, we have found that, with $m = 2$, no instance produced 3, i.e., more than m, order areas. Second: With a rise in m, fixed costs also rise, as the TBO is a multiple of m. This lets the probability of ordering at all decrease, while, at the same time, it makes the absolute spread between the fixed costs of K and L larger. Thus, ordering in the regular order period becomes more beneficial. Another effect is that, for constant fixed costs, the probability of ordering in the regular order period decreases with a higher m, which is simply due to the longer time period between regular shipments. Due to these trade-offs, we hence see a non-monotonic effect of m.

The effect of TBO can be explained by its relation to the fixed costs. As TBO rises, both fixed costs K and L rise and hence the probability of ordering at all decreases. Thus, the importance of the multiple order areas diminishes.

As expected, rising values for K/L lead to smaller cost differences in tendency, as the multiple order areas in the regular order period become less and less important due to the decreasing probability of ordering in the regular order period.

The effect of c_H is not as straight-forward and, as pointed out before in this subsection, depends on the combination with the value for parameter μ. We find that the maximum deviation for $\mu = 5$ always occurs in instances with a higher c_H than for $\mu = 15$ and over all instances the effect of c_H is non-monotonic.

These effects of c_H become clear when we recapitulate our intuition regarding multiple order areas from the end of Section 2.3. Assuming a certain c_H, we might have a policy where it is optimal to always order in the regular order period and cover the demand of all intraperiods to reduce the probability of ordering in the intraperiod to almost zero. Now, if we increase c_H, we might get to the point where this is not longer the case and the trade-off between additional holding costs and the savings in fix order costs varies from one inventory state to the next. In that case multiple order areas might exists. Increasing c_H further will eventually lead to the case where forward-buying in the regular order period is not longer beneficial and again only one order area exists. The same reasoning can be applied to a rise in μ as it is directly connected to a rise in the expected holding costs, assuming that the order policy wants to reduce the probability of ordering in the intraperiods. This shows that the effect of multiple order areas is also sensitive to the combination of both parameters, which makes its impact on costs hard to predict.

Another observation is that optimal values for s_j and S_j in the intraperiods do not monotonically decrease or increase within an order cycle. As an example see Figure 2.2 with parameters $m = 10$, $c_H = 1$, $c_P = L = 260$, $K = 114$, discretized gamma distribution with $(\mu = 20, \sigma = 1)$.

2.5.3.3. Results of the (s, S, r, R), (s, S) and $(Opt|m)$ Policies

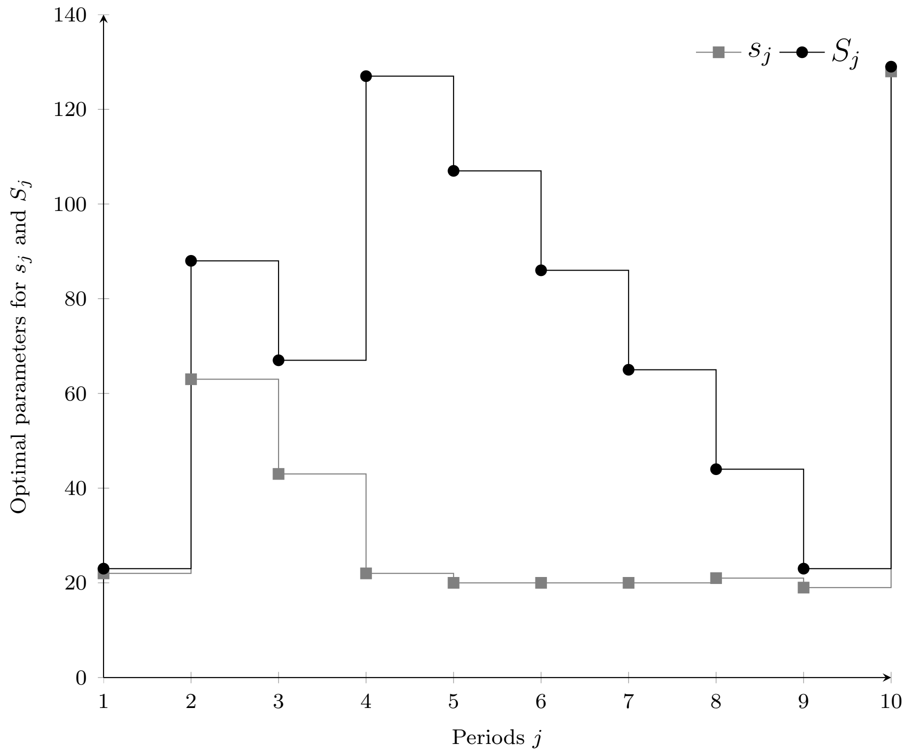
In Table 2.3, we depict the cost differences to the optimal policy in the infinite horizon case for all parameters and the benchmark policies (s, S, r, R), (s, S) and $(Opt|m)$. Again, values show the average over all those instances where the respective parameter value applies.

We first investigate the overall performance of the three policies. It becomes evident that in most instances, the (s, S) policy is not capable of producing good results, with a 32.27 % cost increase on average. The cost deviation rises to a maximum of up to 446.68 % for $\mu = 15$, $CV = 0.1$, $m = 7$, $TBO = 5m$, $K/L = 0.1$ and $c_H = 0.1$. On the other hand, the (s, S, r, R) policy mitigates the cost difference much better with only

Table 2.3.: Average and maximum cost deviations (in %) of (s, S, r, R), (s, S) and $(Opt|m)$ against optimal policy (infinite horizon)

		(s, S)		(s, S, r, R)		$(Opt\|m)$	
Parameter	Value	Avg.	Max.	Avg.	Max.	Avg.	Max.
μ	5	31.89	418.68	0.96	73.86	8.79	86.18
	15	32.65	446.68	0.84	12.82	8.16	73.90
CV	0.1	42.59	446.68	0.32	73.86	8.16	86.18
	0.3	30.77	273.76	1.16	13.93	7.77	47.93
	0.5	23.45	209.14	1.23	9.18	9.50	54.11
m	2	23.82	334.83	0.00	0.00	9.85	86.18
	3	30.19	415.04	0.45	3.42	8.18	47.04
	5	35.93	425.14	1.41	73.86	8.10	51.65
	7	39.14	446.68	1.76	29.04	7.79	54.11
TBO	$0.5m$	3.64	22.90	1.01	6.07	35.33	86.18
	$1m$	20.26	83.75	1.56	73.86	2.97	16.93
	$1.5m$	32.90	143.98	0.73	7.44	1.74	17.14
	$2m$	39.63	195.59	0.69	6.35	1.64	20.27
	$5m$	64.92	446.68	0.52	29.04	0.70	18.25
K/L	0.1	101.71	446.68	1.43	73.86	8.66	86.18
	0.2	65.23	224.80	1.17	8.42	8.36	81.46
	0.3	44.61	132.95	1.09	17.56	8.11	77.23
	0.4	30.87	82.73	1.06	29.04	7.91	73.41
	0.5	20.96	58.38	0.91	13.93	7.81	69.96
	0.6	13.62	40.16	0.95	11.91	7.86	66.82
	0.7	8.06	26.01	0.81	7.21	8.20	63.94
	0.8	4.01	15.30	0.50	4.13	8.98	61.30
	0.9	1.36	6.99	0.20	1.95	10.42	58.88
c_H (c_P)	0.1 (9.9)	29.67	446.68	0.99	13.93	11.66	73.90
	0.5 (9.5)	31.82	387.68	1.02	73.86	8.71	70.45
	1 (9)	33.27	335.00	0.89	29.04	7.51	86.18
	2 (8)	34.31	264.22	0.70	9.18	6.03	78.89
Total		32.27	446.68	0.90	73.86	8.48	86.18

Figure 2.2.: Example values for s_j, S_j for each period of the order cycle



0.90 % on average. However, costs can still be off by up to 73.86 % for $\mu = 5$, $CV = 0.1$, $m = 5$, $TBO = 1m$, $K/L = 0.1$ and $c_H = 0.5$. For the policy where only ordering in the regular order period is allowed ($Opt|m$), we observe that on average costs are 8.48 % higher than the optimal solution and worse by a maximum of 86.18 % for $\mu = 5$, $CV = 0.1$, $m = 2$, $TBO = 0.5m$, $K/L = 0.1$ and $c_H = 1$.

Similar to our findings of Section 2.5.3.2, maximum cost deviations for the three policies occur when CV and K/L are at their lowest value. Another similarity is that either large mean demand and small holding costs or small mean demand and larger holding costs produce the biggest cost differences. However, m and TBO values for the instances with the biggest cost differences do not show identical results for the models. For the (s, S) policy, large values for TBO and m produce the worst solutions. For (s, S, r, R), the biggest cost difference can be found at medium levels for both m and the TBO. Lastly, ($Opt|m$) performs worst when m and TBO are very small. This comparison already indicates that the all three benchmarks have weaknesses (and strengths) in different parameter settings, all of which we explore in the following.

The mean demand has only a negligible effect on the average performance of all three models.

An interesting insight is that, with an increased volatility in the demand, i.e., a larger CV, the differences between the (s, S) and the optimal policy decreases. This is due to higher safety stocks and thus higher holding costs, which diminish the effect of saving fixed costs when engaging in a forward-buying in the regular order period.

For the (s, S, r, R) policy, cost differences to the optimal policy are primarily driven by the constant values for (s, S) throughout the intraperiods. In Table 2.3, we see that average cost deviations increase for larger CV. Our observations on the probabilities of ordering (Section 2.5.3.1) show in Table 2.1 that the probability of ordering in the intraperiod is higher for a higher CV. These effects directly connect to the behavior of the (s, S, r, R) policy as any misalignment in the values for (s, S), due to their constant nature over time, leads to a penalization that rises with the probability of ordering in the intraperiods.

We observe a non-monotonic behaviour for rising values in CV and K/L regarding the $(Opt|m)$ policy. These can be explained in context with the TBO values, as depicted in Table 2.4.

For the smallest value of $TBO = 0.5m$, the cost differences that can be observed are by far the greatest. In this case, fixed costs are very low so that holding inventories is comparably expensive in relation to the fixed costs. Hence, ordering in the intraperiod is beneficial. We see that, for $TBO = 0.5m$, the smallest CV leads to the biggest cost differences, while for larger TBO values, the smallest CV leads to the smallest cost differences. Similarly, for $TBO = 0.5m$, cost differences decrease monotonically with a rise in K/L, while, for $TBO > 1m$, cost differences rise with a rise in K/L.

To explain these observations, we need to distinguish between the forward-buying and the batch ordering effect. The forward-buying effect aims at *reducing the ordering probability in the intraperiod*, while at the same time increasing the ordering probability in the regular order period. The batch ordering effect aims at *reducing the probability of ordering at all*, i.e. in both types of periods. Hence, to realize potential savings based on the fixed costs difference, the first effect tries to achieve that orders occur in the regular instead of the intraperiod. As opposed to this, the second effect occurs irrespective of a difference in the fixed costs.

Table 2.4.: Average cost deviations of $(Opt|m)$ against optimal policy (infinite horizon) for varying values of TBO, CV and K/L (in %)

		TBO				
Parameter	Value	$0.5m$	$1m$	$1.5m$	$2m$	$5m$
CV	0.1	41.34	0.30	0.13	0.55	0.46
	0.3	32.37	2.21	1.34	1.94	1.01
	0.5	34.58	6.63	3.84	2.51	0.68
K/L	0.1	39.84	3.09	0.87	0.27	0.01
	0.2	38.61	2.90	0.78	0.24	0.02
	0.3	37.51	2.74	0.74	0.25	0.07
	0.4	36.30	2.63	0.76	0.32	0.13
	0.5	35.45	2.63	0.85	0.63	0.25
	0.6	34.55	2.74	1.07	1.22	0.44
	0.7	33.90	3.05	1.71	2.25	0.79
	0.8	33.42	3.62	3.27	3.71	1.60
	0.9	33.12	4.56	5.97	6.11	3.19

For the very small TBO values, the absolute cost difference between K and L is also very small. This leads to the point where the forward-buying effect is almost not-existent and dominated by the batch ordering effect. When K/L increases, the average fixed order costs increase, and hence the batch ordering effect reduces the probability of ordering at all. The reduced probability of ordering at all leads to the case where in the intraperiod, too, orders occur less frequently, and hence the cost differences for the $(Opt|m)$ policy decrease.

For larger TBO values, the forward-buying effect gains much more importance. For small K/L ratios, orders hence occur primarily in the regular order period, while, for larger K/L, orders in the optimal policy occur more often in the intraperiods, which increases the cost difference for the $(Opt|m)$ policy. As we can see in the table, this effect is more pronounced than the batch ordering effect.

We argue similarly for the effect of the values for CV in combination with TBO. As we have seen in Table 2.1 for the optimal policy, a rise in CV leads to a reduction in the overall order probability and an increase in the order probability of the intraperiod. As explained in Subsection 2.5.3.1, the first effect is independent of and the second effect is dependent on the fixed costs difference between K and L. Table 2.4 shows that, for the smallest TBO value, the first effect of a reduced probability of ordering at all with a larger CV is more dominant than the increase in the intraperiod order probability. This can be explained by the very small fixed costs values in the case of small TBO values. Again, the relative dominance of the effects is reversed for larger TBO values, where the fixed order costs increase.

A rise in m leads to an increase in the average cost deviation of the (s, S) and (s, S, r, R) policies. For the (s, S) policy, the increase in the level of fixed costs, and in turn the absolute fixed costs difference between K and L, is the main driver for performance. For the (s, S, r, R) policy, a larger value for m means a higher probability of ordering in the intraperiods relative to the probability of ordering in the regular order period. Hence, the probability of costs associated with the restricted constant order policy there is higher. A small TBO, on the other hand, leads to a higher probability of ordering in the intraperiods, for which the policy is not fully optimized. For the $(Opt|m)$ policy, a larger order cycle m reduces the cost disadvantages. Larger m and larger TBO both lead to higher absolute cost differences between K and L, which makes it more attractive to order in the regular order period and thus reduces the negative consequences of not being able to order in the intraperiods.

An intuitive observation for the (s, S) and (s, S, r, R) heuristics is the convergence between the heuristics and the optimal policy for $\frac{K}{L} \rightarrow 1$, i.e. when the fixed costs differences become negligible.

Lastly, c_H clearly worsens the performance of (s, S) and improves the performance of $(Opt|m)$, whereas the effect on (s, S, r, R) is as ambiguous as for the (s_j, S_j) policy. The effect on the (s, S) policy can be explained by the induced effect of forward-buying for smaller c_H. If forward-buying is favorable, it will be executed in the low fixed order costs period and lead to much higher order quantities than in the other periods. However, the two constant parameters of the (s, S) policy cannot cover the effect and hence the policy performs worse. $(Opt|m)$ performs in the exact opposite direction, as forward-buying in the regular order period is the only possibility to cover the demand for the intraperiods in this approach. For the effects on the (s, S, r, R) policy, see the explanations of the similar observations regarding the (s_j, S_j) policy at the end of Subsection 2.5.3.2.

2.6. Conclusion

In this chapter, we investigated inventory control policies where, at regular order periods, orders can be placed at less fixed costs than in the intraperiods in between. We proved the optimality of an (s_j, S_j) inventory policy for the intraperiods and of a more complex inventory policy with multiple order areas and order-up-to levels at the regular order periods. We introduced several benchmark policies and (modified) policy iteration algorithms to find optimal policy parameters. A numerical study showed that, despite the potential complex structure in the regular order period, cost differences to an (s, S, r, R) policy are small (0.9 % on average) and nearly non-existent for (s_j, S_j) policies (< 0.01 % on average). The cost differences between benchmarks and the optimal policy reveal a forward-buying effect. The effect aims at redirecting orders from the intra- to the regular order period.

The optimal policy in the regular order period, although complex, can be described by multiple order areas. Comparing it to a policy that potentially necessitates a different order logic for every inventory state, we showed that the order policy contains a certain structure, which allows a rule-based decision-making for managers. More importantly, the intraperiods still follow the well-known (s_j, S_j) policy, with a reorder point and an order-up-to level for each period. Hence, managers may use the concept for following the optimal policy for the intraperiods and use it to approximate the optimal policy in the regular order period.

Further insights for practice are provided by our comparison of the benchmark policies. Neither the (s, S) policy with the same parameters for all periods, nor the $(Opt|m)$ policy, which orders only in the cheaper fixed order costs period, can be recommended because both show large cost differences to the optimal policy. However, with as few as four parameters, the (s, S, r, R) policy already mitigates a large share of these cost differences in most of the instances. Still, if fixed costs differences, the CV, the length of the order cycle and the time between orders are large, then cost differences might be substantial and hence an (s_j, S_j) policy should be introduced.

For further research, using other sub-optimal inventory control strategies might also be worthwhile to investigate. Due to the different fixed costs, we observe that we have a forward-buying effect in the regular order period, as we preferably do not want to order in an intraperiod. This situation has some similarities with the speculation environment of Gavirneni and Morton (1999). Hence, their approaches might be altered and applied to our setting to provide additional benchmark policies.

The algorithms we used in this chapter rely on traditional policy iteration mechanisms and hence suffer from the curse of dimensionality. To achieve better computational performance, other algorithms, e.g., based on the renewal theory, should be investigated.

2.A. Appendix: Notion of K-convexity

We provide some well-known definitions and properties on K-convexity, which we need for the construction of the optimality proofs.

We start with a definition of K-convexity based on Sethi and Cheng (1997):

Definition 2.2. *A function $g : R \to R$ is said to be K-convex if $K \geq 0$, and it satisfies the property,*

$$K + g(z + y) \geq g(y) - z\frac{g(y) - g(y - b)}{b}, \forall z \geq 0, b > 0, y \tag{2.7}$$

.

Furthermore, we use some important propositions of K-convexity (Sethi & Cheng, 1997).

Proposition 2.3. *(i) If $g : R \to R$ is K-convex, it is L-convex for any $L \geq K$. In particular, if g is convex, i.e. 0-convex, it is also K-convex for any $K \geq 0$.*

(ii) If g_1 is K-convex and g_2 is L-convex, then for $\alpha, \beta \geq 0$,$\alpha g_1 + \beta g_2$ is $(\alpha K + \beta L)$-convex.

(iii) If g is K-convex and ξ is a random variable such that $E|g(x - \xi)| < \infty$, then $Eg(x - \xi)$ is also K-convex.

(iv) If g is K-convex, $x < y$ and $g(x) = K + g(y)$, then $g(z) \leq K + g(y)$ for all $z \in [x, y]$.

Proof. For the proof of (i)-(iii) see Sethi and Cheng (1997) and for (iv) see Scarf (1959). □

Proposition 2.4. *(Sethi & Cheng, 1997) Let $g : R \to R$ be a K-convex function such that $g(x) \to +\infty$ as $x \to +\infty$. Let the notation $g(-\infty)$ denote the extended real number $\liminf_{x \to -\infty} g(x)$. Let*

$$g^* = \inf_{x \in (-\infty, \infty)} g(x) > -\infty \tag{2.8}$$

Define the extended real numbers S and s, $S \geq s \geq -\infty$ as follows:

$$S = \min \{x \in R \cup -\infty | g(x) = g^*, -\infty < x < \infty\}, \tag{2.9}$$
$$s = \min \{x \in R \cup -\infty | g(x) \leq K + g(S), -\infty < x \leq S\}. \tag{2.10}$$

Then

(i) $g(x) \geq g(S), \forall x \in (-\infty, \infty)$;

(ii) $g(x) \leq g(y) + K, \forall x, y$ *with* $s \leq x \leq y < \infty$

(iii) $h(x) \equiv \inf_{y \geq x, -\infty < y < \infty} [K\delta(y-x) + g(y)] = \begin{cases} K + g(S) & \text{for } x < s \\ g(x) & \text{for } s \leq x < \infty \end{cases}$ *and if g is continuous then* $h : R \to R$ *is continuous;*

(iv) h *is K-convex on* $(-\infty, \infty)$

Moreover, if $s > -\infty$*, then*

(v) $g(s) = K + g(S)$;

(vi) $g(x)$ *is strictly decreasing on* $(\infty, s]$.

Proof. See Sethi and Cheng (1997). □

2.B. Appendix: Proofs

2.B.1. Proof of Proposition 2.1

2.B.1.1. Proof of (ii)

We prove (ii). If $\bar{A} = -\infty$, then $h(x) = g(x)$ and this is obviously true. Next if $\bar{A} > -\infty$ and $\underline{A} > -\infty$, we need to verify

$$L + h(y+z) - \left[h(y) + z\frac{h(y) - h(y-b)}{b}\right] \geq 0 \tag{2.11}$$

in the following cases:

(a) $y-b \in (\underline{a}_i, \bar{a}_i) : i \in I$ and $y \in (\underline{a}_k, \bar{a}_k) : k \in I, k \geq i$ and $y+z \in (\underline{a}_l, \bar{a}_l) : l \in I$, $l \geq k$

(b) $y-b \in (\underline{a}_i, \bar{a}_i) : i \in I$ and $y \notin (\underline{a}_k, \bar{a}_k) \forall k \in I, k \geq i$ and $y+z \in (\underline{a}_l, \bar{a}_l) : l \in I$, $l \geq j$

(c) $y-b \in (\underline{a}_i, \bar{a}_i) : i \in I$ and $y \in (\underline{a}_k, \bar{a}_k) : k \in I, k \geq i$ and $y+z \notin (\underline{a}_l, \bar{a}_l) \forall l \in I$, $l \geq k$

(d) $y-b \in (\underline{a}_i, \bar{a}_i) : i \in I$ and $y \notin (\underline{a}_k, \bar{a}_k) \forall k \in I, k \geq i$ and $y+z \notin (\underline{a}_l, \bar{a}_l) \forall l \in I$, $l \geq j$

(e) $y-b \notin (\underline{a}_i, \bar{a}_i) \forall i \in I$ and $y \in (\underline{a}_k, \bar{a}_k) : k \in I$ and $y+z \in (\underline{a}_l, \bar{a}_l) : l \in I$, $l \geq k$

(f) $y-b \notin (\underline{a}_i, \bar{a}_i) \forall i \in I$ and $y \in (\underline{a}_k, \bar{a}_k) : k \in I$ and $y+z \notin (\underline{a}_l, \bar{a}_l) \forall l \in I$, $l \geq k$

(g) $y-b \notin (\underline{a}_i, \bar{a}_i) \forall i \in I$ and $y \notin (\underline{a}_k, \bar{a}_k) \forall k \in I$ and $y+z \in (\underline{a}_l, \bar{a}_l) : l \in I$

(h) $y-b \notin (\underline{a}_i, \bar{a}_i) \forall i \in I$ and $y \notin (\underline{a}_k, \bar{a}_k) \forall k \in I$ and $y+z \notin (\underline{a}_l, \bar{a}_l) \forall l \in I$

Case (a): For proving (a) we can rewrite the inequality (2.11) to

$$L + g(a_l^*) - g(a_k^*) - z\frac{g(a_k^*) - g(a_i^*)}{b} \geq 0$$

and have to check the following sub-cases: (a1) $a_i^* = a_k^* = a_l^*$, (a2) $a_i^* < a_k^* = a_l^*$, (a3) $a_i^* < a_k^* < a_l^*$ and (a4) $a_i^* = a_k^* < a_l^*$. In (a1) inequality (2.11) simplifies to $L \geq 0$, which is true by definition. In (a2) and (a3) we can use $b \geq y - a_i^*$, as $y - b \leq a_i^*$ from Definition 2.1.(iii). L-convexity of g yields

$$L + g(a_l^*) - \left[g(a_k^*) + z\frac{g(a_k^*) - g(a_i^*)}{b}\right] \geq L + g(a_l^*) - \left[g(a_k^*) + z\frac{g(a_k^*) - g(a_i^*)}{y - a_i^*}\right] \geq 0.$$

(a4) simplifies to $L + g(a_l^*) - g(a_k^*) \geq 0$, which has to be true as $g(a_l^*) \geq g(a_k^*)$ based on a_j^* in Definition 2.1.(iii).

Case (b): Next, we prove (b), where we can rewrite the inequality (2.11) to

$$
\begin{aligned}
L + K + g(a_l^*) &- \left[g(y) + z\frac{g(y) - (K + g(a_i^*))}{b}\right] \\
= L + K + zK + g(a_l^*) &- \left[g(y) + z\frac{g(y) - g(a_i^*)}{b}\right] \\
\geq L + g(a_l^*) &- \left[g(y) + z\frac{g(y) - g(a_i^*)}{b}\right] \\
\geq L + g(a_l^*) &- \left[g(y) + z\frac{g(y) - g(a_i^*)}{y - a_i^*}\right] \geq 0.
\end{aligned}
$$

In the last step of the inequality, we apply the same reasoning as for (a2) and (a3).

Case (c): In (c) the inequality becomes

$$
\begin{aligned}
L + g(y+z) &- \left[g(a_k^*) + K + z\frac{g(a_k^*) - g(a_i^*)}{b}\right] \\
= (L - K) + g(y+z) &- \left[g(a_k^*) + z\frac{g(a_k^*) - g(a_i^*)}{b}\right] \geq 0
\end{aligned}
$$

and we distinguish the cases, where (c1) $a_i^* = a_k^*$ and (c2) $a_i^* < a_k^*$. In (c1), we get $(L-K) + (g(y+z) - g(a_k^*)) \geq 0$ where obviously $(L-K) \geq 0$ and $(g(y+z) - g(a_k^*)) \geq 0$ following that $g(y+z) > g(a_k^*)$ as a_k^* is the minimizer on $[y, \infty]$ and $y < y + z < \infty$. In (c2), we can show that as $g(a_k^*) + K \leq g(y)$ and $y - b \leq a_i^*$ it holds that $b \geq y - a_i^*$

$$
\begin{aligned}
L + g(y+z) - \left[g(a_k^*) + K + z\frac{g(a_k^*) - g(a_i^*)}{b}\right] &\geq L + g(y+z) - \left[g(y) + z\frac{g(a_k^*) - g(a_i^*)}{b}\right] \\
\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(a_i^*)}{b}\right] &\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(a_i^*)}{y - a_i^*}\right] \geq 0
\end{aligned}
$$

and use L-convexity of g to conclude the proof.

Case (d): In (d) the inequality can be proven like in (b), as

$$
\begin{aligned}
&L + g(y+z) - \left[g(y) + z\frac{g(y) - (g(a_i^*) + K)}{b}\right] \\
&= L + zK + g(y+z) - \left[g(y) + z\frac{g(y) - g(a_i^*)}{b}\right] \\
&\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(a_i^*)}{b}\right] \\
&\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(a_i^*)}{y - a_i^*}\right] \geq 0.
\end{aligned}
$$

Case (e): In (e), we use the property that $g(y) \geq g(a_k^*) + K$. As $a_l^* \geq y + z$ it holds that $z \leq a_l^* - y$. We thus come to the following:

$$
\begin{aligned}
&L + g(a_l^*) + K - \left[g(a_k^*) + K + z\frac{g(a_k^*) + K - g(y-b)}{b}\right] \\
&\geq L + g(a_l^*) - \left[g(y) + z\frac{g(a_k^*) + K - g(y-b)}{b}\right] \\
&\geq L + g(a_l^*) - \left[g(y) + z\frac{g(y) - g(y-b)}{b}\right] \\
&\geq L + g(a_l^*) - \left[g(y) + (a_l^* - y)\frac{g(y) - g(y-b)}{b}\right] \geq 0
\end{aligned}
$$

where we again can prove the last inequality by L-convexity of g.

Case (f): In (f), we follow the same as in (e), with using $g(y) \geq g(a_k^*) + K$ to show that

$$
\begin{aligned}
&L + g(y+z) - \left[g(a_k^*) + K + z\frac{g(a_k^*) + K - g(y-b)}{b}\right] \\
&\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(y-b)}{b}\right] \geq 0
\end{aligned}
$$

and prove the last inequality by L-convexity of g.

Case (g): In (g), we can prove as in (b) that

$$L + g(a_l^*) + K - \left[g(y) + z\frac{g(y) - g(y-b)}{b}\right] \geq L + g(a_l^*) - \left[g(y) + z\frac{g(y) - g(y-b)}{b}\right]$$
$$\geq L + g(a_l^*) - \left[g(y) + z\frac{g(y) - g(y-b)}{y-b}\right] \geq 0.$$

Case (h): In (h), we can prove as in (b) that

$$L + g(y+z) - \left[g(y) + z\frac{g(y) - g(y-b)}{b}\right]$$
$$\geq L + g(y+z) - \left[g(y) + z\frac{g(y) - g(y-b)}{y-b}\right] \geq 0.$$

2.B.1.2. Proof of (i), (iii), (iv)

The proofs of (iii) and (iv) directly follow from Proposition 2.4.(iii) as g is L-convex.

The proof of (i) is as follows. Assuming that $\liminf_{x\to\infty} g(x) > g(M) + K$, we can use Definition 2.1 by proving all possible outcomes of the right hand side of the equation. We will start to prove that $h(x) = K + g(M)$ for $x < \underline{A}$. To do so, we consider $\bar{a}_0$. For $\underline{a}_0 < x < \bar{a}_0 = \underline{A}$ it holds that $g(x) > g(a_0^*) + K$, which is why ordering up to a_0^* is optimal. As $\underline{a}_0 = -\infty$, $a_0^* = \arg \inf_{x\in(-\infty,\infty)} g(x) = M$, it is always optimal to order up to M in this case.

The second outcome, that $h(x) = K + g(a_i^*)$ for all $i \in I, x \in (\underline{a}_i, \bar{a}_i)$, directly follows from the Definition 2.1.(iv).

To prove that $h(x) = g(x)$ for all $i \in I, x \in [\bar{a}_{i-1}, \underline{a}_i]$, we check for the cases where (a) $\bar{a}_{i-1} \leq x \leq a_{i-1}^* \leq \underline{a}_i$, (b) $\bar{a}_{i-1} \leq x \leq \underline{a}_i \leq a_i^*$ and (c) $\bar{a}_{i-1} \leq a_{i-1}^* \leq x \leq \underline{a}_i \leq a_i^*$.

In ((a)), it holds that $g(x) \leq g(a_{i-1}^*) + K$ which is why it is optimal not to order, as a_{i-1}^* is the global minimum on $[x, \infty)$ and the above inequality still holds.

((b)) follows from ((a)) as $a_{i-1}^* = a_i^*$ in this case.

In ((c)), we have to compare with the global minimum $y^* = \arg \inf_{y\in(x,\infty]} g(y)$. If $y^* < a_i^*$, either $g(x) \leq g(y^*) + K$ and it is optimal not to order or $g(x) > g(y^*) + K$, which by the Definition 2.1.(ii) would mean $x > \underline{a}_i$, which is in contradiction to $x \in [\bar{a}_{i-1}, \underline{a}_i]$.

To prove the last outcome of $h(x) = g(x)$ for $\bar{A} \leq x \leq \infty$, we can again compare with the global minimum $y^* = \arg \inf_{y\in(x,\infty]} g(y)$. If $g(x) \geq g(y^*)+K$, then there is another $\underline{a}_{|I|+1}$

and $\bar{a}_{|I|+1}$ by Definition 2.1.(ii) and 2.1.(iv) and thus $\bar{A} > x$, which is in contradiction to $\bar{A} \leq x \leq \infty$. We will next prove that it is also not optimal to order up to any other point $z \neq y^*, z > x$ by contradiction. Assume it is optimal to order, then $g(x) > g(z) + K$. As y^* is the global minimum, it holds that $g(z) \geq g(y^*)$ and therefore $g(x) \geq g(y^*) + K$, which cannot be the case as shown before.

It remains to show that $h(x)$ is continuous. As $g(x)$ is continuous and $g(x) \leq K + g(a_i^*)$ for $i \in I$, $x \in \{\underline{a}_i, \bar{a}_i\}$, $h : R \rightarrow R$ is also continuous.

This completes the proof. □

2.B.2. Proof of Theorem 2.1

First, we will prove that $G_{n,j}$ is a continuous and L-convex function for all $j = 1, \ldots, m$. Assuming that $f_{n,j+1}(x)$ and $f_{n+1,1}(x)$ are continuous and L-convex, $G_{n,j}$ is a sum of two convex functions and a third term $\int_0^\infty f_{n,j+1}(y - \xi)\phi(\xi)d(\xi)$ for all $j = 1, \ldots, m-1$ or a third term $\int_0^\infty f_{n+1,j}(y - \xi)\phi(\xi)d(\xi)$ for $j = m$, which are L-convex by Proposition 2.3.(iii) and 2.3.(ii). Therefore, $G_{n,j}$ is also L-convex, by Proposition 2.3.(ii) and Proposition 2.3.(i).

By Proposition 2.4.(iv), $f_{n,j}$ is also L-convex and continuous for all $j = 1, \ldots, m-1$. Note that Proposition 2.4.(iv) does not hold for $f_{n,m}$. However, Proposition 2.1.(ii) for $f_{n,m}$ proves L-convexity.

Iterating backwards, starting at $f_{N+1,1} = f_{TE}$, which is continuous and by definition L-convex. As $G_{n,j}$ is L-convex for all $j = 1, \ldots, m$ and it is furthermore straightforward to show that $G_{n,j}$ is continuous, we can apply Proposition 2.1.(i) to obtain the optimal policy for the regular order period $j = m$ and apply Proposition 2.1.(iii) to get the optimal policy for the intraperiods $j = 1, \ldots, m-1$. □

Chapter 3.

Data-driven assignment of delivery patterns with handling effort considerations in retail

We consider a supply chain with one warehouse and multiple stores. At the warehouse, the orders for the stores are picked and in the store, shelves are stacked from the backroom. We include handling costs at the warehouse and stores as these are main drivers for logistics costs. We find delivery patterns and order-up-to levels, both of which shall remain fixed for a certain time. As especially in retail stochastic non-stationary demand structures are prevalent, we extend the classic joint replenishment problem under dynamic demand by a stochastic yet distribution-free optimization approach based on historical data samples. We formulate a mixed integer linear program using the plant-location formulation and develop several hierarchical decomposition approaches and a genetic algorithm. We consider a cyclic approach for orders, which allows an order at the end of the time horizon to fulfill the demand at the beginning of the time horizon. Using this approach, there is no need for initial inventories to be set as an input; they are are optimized within the model. Furthermore, a metacalibration approach is introduced, which allows an automated setting of input parameters for the genetic algorithm. To derive insights into the performance of the models, random instances are solved and then the most promising models are used for a case study with a European retailer. The results for the controlled test instances are analyzed by a meta-modeling approach that provides insights into performance drivers for the investigated model variants. The average logistics cost savings of our model over a deterministic approach with safety stocks amount to 3.02 % for the controlled test instances. In a similar comparison for the case study, average results over different parameter combinations show a 20.60 % logistics costs saving potential.

3.1. Introduction

The problem we deal with reflects a common situation at retail chains that face inventory control decisions across many individual stores and a large amount of products, see for example Sternbeck and Kuhn (2014). These companies set fixed delivery patterns for each store and product valid for several weeks, which are revised on a rolling basis. A delivery pattern determines during which periods a delivery from the warehouse to a store is made (e.g. on Monday morning and Wednesday afternoon or only on Thursday morning). Associated with the fixed delivery patterns, order-up-to levels are defined and fixed in the same way. Although this fixed inventory control policy establishes a stable setting for operations, demand in retail is stochastic and clearly non-stationary (Martel et al., 1995).

The model presented in our work includes a data-driven (or robust, distribution-free) approach (see Bertsimas & Thiele, 2014). Historical data points used for the calculation represent multiple samples of the target time horizon (e.g. several weeks). The decisions made in the model are applied to all sample weeks in the same way, e.g. if a delivery on Monday morning with an order-up-to level of 10 is scheduled, a delivery is carried out in all sample weeks where the inventory level is less than 10. This approach has been developed by Iyer and Schrage (1992) to optimize a deterministic (s,S) policy. Bertsimas and Thiele (2006) solve inventory planning problems with a data-driven linear programming formulation and Beutel and Minner (2012) use the approach to determine safety stocks. Other applications include, e.g., the data-driven minimization of emissions in urban areas of Ehmke et al. (2016). If this method is used, parametric distributions do not have to be fitted to sales data. Instead, this data is used as a direct input for optimization. By applying this method, we circumvent the estimation of suitable parameters and choice of theoretical distributions. The approach, therefore, does not need specific knowledge about a suitable parametric distribution (and not even about the moments of the distribution as in, e.g., the distribution-free approach of Braglia et al., 2017). As inputs are used as-is, the model is easily interpretable for managers. Furthermore, the modeling approach based on a mixed integer linear program (MILP), allows practical constraints, e.g., the addition of capacity constraints to the model. Using the data-driven approach, we rely on demand data as inflated sales, i.e. data that has not been censored by out-of-stock situations. Especially in retail, it has been discussed how to gain this data. For a distribution-free approach, see Sachs and Minner (2014).

Due to the non-stationarity, we get varying utilizations of warehouses and stores from one period to the next. Capacities in terms of available labor at the warehouse and store are scarce as only a limited number of employees is available. Therefore, we incorporate capacity restrictions for the total effort in each period. In doing so, we achieve a smoothing of the load throughout the periods, as, due to the store clerks' shift schedules, store managers only have limited flexibility for coping with high workload peaks. Neglecting this constraint will most likely lead to a high accumulation of delivery patterns right before the weekend, where demand is the highest, which might not be manageable by the workforce in the warehouse. Note that, in a practical setting, these capacity constraints might be soft, which is why managers who use the model should perform a parameter variation on the available capacity to observe the most suitable trade-off between a balanced load and costs. The capacity constraint also serves as an example of how a practical constraint can be included into the MILP formulation.

Given the high percentage of manual labor in a retailer's logistics costs (38 % store handling and 28 % warehouse handling Van Zelst et al., 2009), this important cost factor has to be included in the decision making. The necessity to include handling effort in the decision making of retailers has been shown empirically by Van Zelst et al. (2009) and Reiner et al. (2013). DeHoratius and Ton (2009) conclude that backrooms in stores are often organized poorly and shelf stacking tasks are tedious for store clerks. Van Donselaar et al. (2010) find that order systems neglecting handling efforts are overruled by the store workers anyway. Curşeu et al. (2008) extend the economic order quantity model by adding a handling cost factor and investigate the structure of handling efforts at the store via regression analysis. This analysis supports that there is an additive structure of fixed and variable terms for explaining the handling effort.

Van Woensel et al. (2013) model the problem, including handling efforts as a Markov decision process, and numerically determine the complex long-term cost optimal policy, as well as the performance of structures like (s, S), (s, Q) and a mixture of both. However, orders can be placed during any period, i.e. there is no fixed order (and thus delivery) pattern assigned to the stores and a stationary theoretical demand distribution is assumed. Focusing on the handling effort in a warehouse in a stochastic demand setting, Kiesmüller and Broekmeulen (2010) find that including joint replenishment effects of the warehouse into the decision-making process leads to significant cost savings. They, however, assume that no joint replenishment effects occur at the downstream node (in our case stores).

So far, there has only been a limited number of publications dealing with the handling effort in retail logistics in general and with a quantitative implementation specifically. This is especially true when we are looking at the determination and assignment of delivery patterns for stores with non-stationary demand. The contributions of this chapter include an application of the data-driven solution approach for the joint replenishment problem (JRP) with stochastic non-stationary demand, the introduction and comparison of several hierarchical decomposition methods and a savings heuristic, a cyclical approach for including decisions on the initial inventory into the decision making process and the introduction of a meta-calibration method for a genetic algorithm (GA).

The chapter is organized as follows: After a brief literature review in Section 3.2, the model formulation, and the mathematical program are depicted in Section 3.3. Hierarchical decomposition and solution methods are covered in Section 3.4. Benchmark models are introduced in Section 3.5. A numerical study based on fictitious data is presented and the most promising models are then applied to real data of a European retailer in Section 3.6. Concluding remarks and ideas for further research are given in Section 3.7.

3.2. Literature Review

Modeling the deliveries from the warehouse to the stores, along with modeling the shelf stacking in the store are key parts of the problem. This modeling resembles the joint replenishment problem. The key to the problem is to decide when it is optimal to place a combined order for multiple products, although it might not be optimal for each individual product. For an overview of relevant publications, see Aksoy and Erenguc (1988) and Khouja and Goyal (2008).

In JRP publications that deal with non-stationary stochastic demand, most solutions are based on the three heuristic strategies of static uncertainty, dynamic uncertainty and static-dynamic uncertainty as established in Bookbinder and Tan (1988). The first strategy implements all decisions taken at the start of the planning horizon, whereas the second strategy assumes that decisions are made in each period and the third strategy assumes that decisions are made for the whole horizon, but only implemented for the first period and then updated in a rolling fashion.

Martel et al. (1995) consider a constrained multi-item non-stationary demand setting and develop a stochastic programming formulation for the static-dynamic uncertainty approach, where the replenishment plan of the whole horizon is revised in each period, which results in multiple static demand sub-problems. A stochastic program with simple recourse for the capacitated lot sizing problem with probabilistic time-varying demand is constructed and an equivalent deterministic program is created. Similarly, Hua et al. (2009), again in a static-dynamic approach, re-solve the problem in each period based on updated demand distributions. It is assumed in both papers that decisions can be changed with every revision of the plan, which means that they do not truly represent robust decisions. As our retail setting, however, calls for the decisions to be robust, their approaches are not applicable. Besides, only major setup costs are considered while minor setup costs are neglected.

Tempelmeier and Hilger (2015) apply the static uncertainty approach, which follows the decision-making process in our problem setting more closely. Setup periods as well as lot sizes are defined at the beginning of the planning horizon and not revised later. The authors formulate a MILP model to solve the stochastic multiple-item capacitated lot-sizing problem with β service level constraints and setup carry-overs. They propose a fix & optimize heuristic, where the problem is decomposed into sub-problems with a smaller number of items. At the end, the best solutions of all sub-problems are merged to get a solution for the whole problem. This is similar to our approach of a product and store based decomposition, except that we decompose for all individual products and merge solutions of individual sub-problems by a selection MILP in a second stage.

The determination of delivery schedules and quantities in multiple locations and echelons closely relates to the inventory routing problem (IRP). A given set of customers has to be served by a central warehouse in a multi-period environment, where the decision when to serve each customer, how much inventory to allocate and how to construct best routes for the trucks in each period must be made (for reviews see Coelho et al., 2013; Moin & Salhi, 2006). The special case of the periodic IRP determines fixed delivery patterns valid throughout the whole time horizon. Campbell and Savelsbergh (2004) use a hierarchical decomposition approach on the IRP where they first construct delivery patterns with a MILP formulation and subsequently create routes heuristically in a second stage. Jaillet et al. (2002) first define optimal replenishment days for each customer based on an equidistant replenishment interval and re-optimize based on the opportunity cost of switching customers to days other than their designated replenishment days. They justify the equidistance of times between replenishments by stating that, for demand with seasonality, the time horizon should be split into smaller parts stationary in their demands. Schedules with no equidistance restrictions in their deliveries and based on the periodic IRP are developed by Gaur and Fisher (2004) in a retail environment. They aggregate the demand over all products to end up with a single-item problem. Also, they neglect the variance in demand during the creation of delivery patterns.

Mixed integer linear programming approaches can be found for the joint replenishment problem under dynamic demand (JRPDD), where deterministic time varying demand structures are taken into account. An overview of different approaches and their comparison can be found in Boctor et al. (2004) and Gao et al. (2008). In their comparisons, the plant-location type formulation of the problem yields the best performances, which is why we, too, use this formulation type. Although a deterministic production setting, the work of Mehrotra et al. (2011) gives interesting insights into using production patterns in the real-world setting of a food processing company with a huge number of products. Applying a MILP formulation and a heuristic, they come up with costs savings of 28 % in the real world case against the solution previously implemented by a consultant. To deal with the complex setting of simultaneous lot-sizing and supplier selection decisions, Cárdenas-Barrón et al. (2015) use a reduce-and-optimize approach. This approach constructs sub-problems with a reduced feasible space, solves them and infers from the reduced costs of additional variables which variables to further include in the decision space for the next iteration.

Several hierarchical decomposition approaches have been developed, especially for the IRP. Raa and Aghezzaf (2009) implement a column generation approach for creating and selecting appropriate delivery patterns. Multiple hierarchical decomposition approaches with sequential decision making are considered by Bertazzi et al. (2005), who tackle a production-distribution system. Cordeau et al. (2014) implement a three-phased hierarchical decomposition in the last phase of which a MILP solves the integrated planning and routing problem.

For the creation of (delivery) patterns, metaheuristics have been widely used in the periodic vehicle routing problem (PVRP), which is related to the periodic IRP but lacks the decision how to allocate inventory to customers. A GA approach for the PVRP in combination with neighborhood-based searches and a new population management has been developed by Vidal et al. (2012). The authors claim that their approach outperforms all others in all the available benchmark classes. Drummond et al. (2001) decompose the PVRP and use a combination of a parallel GA for the creation of delivery patterns and, in a second step, a savings heuristics for route evaluation. In the consistent vehicle routing problem in Groer et al. (2009), several aspects (e.g. driver, time of delivery, etc.) have to remain unchanged for a customer over multiple delivery periods. Kovacs et al. (2014) tackle this by creating so-called template routes that encapsulate solutions for combinations of customers requiring consistency. Based on whether or not those customers have a delivery in a certain period, daily routes are derived. Initial templates are improved by an adaptive large neighborhood search.

In the IRP literature, we find that metaheuristics are scarce (for an overview on metaheuristic approaches, see Coelho et al., 2012). Abdelmaguid and Dessouky (2006) and Moin et al. (2011) use a GA for the deterministic IRP with multiple periods. Abdelmaguid and Dessouky (2006) split the problem by constructing delivery patterns within the GA chromosome and, in a second step, solving the vehicle routing part using heuristics. Moin et al. (2011) optimize both sub-problems simultaneously with a GA approach.

3.3. Data-driven Delivery Pattern Assignment Problem

3.3.1. Assumptions

From a logistics network perspective, our model represents a divergent three-echelon system with one central warehouse serving multiple retail stores and with the backroom serving the shelves on the sales floor within each store. We assume that demand is non-stationary stochastic and use samples of several weeks of data for deterministic time-varying demand points as input for the model. Lead time is zero, assuming that there is enough time for the delivery of orders between two successive periods. We target commonly available products where historic data is available (i.e. no new products) and no systematic influence on the future demand is invoked by the company (e.g. via promotions or rebates). However, both could be incorporated into the approach by an appropriate sampling of the demand data.

3.3.2. Sequence of Events

In the beginning of each period t, orders are placed in each store $s \in S$ and for each product $p \in P$ if a delivery is scheduled in the respective delivery pattern (z_{tps}). z_{tps} is binary and takes the value 1 if a delivery is scheduled in period t for product p in store s. The amount ordered is chosen such that the inventory level is raised to a specified order-up-to level (OUL_{tps}). Picking in the warehouse takes place based on all orders across the stores and products. As soon as picking is finished, the orders are shipped to the different retail stores and arrive instantaneously. Based on what is available in the backroom after the delivery arrives and the inventory available on the shelves, some shelf stacking might take place. The shelf stacking is assumed to be done for multiple products within the same period. After the stacking events, customer demands (D_{nps}) take place in the store up to the point where the shelf space of that product (SH_{ps}) is empty. In our MILP approach, decision variables are defined as fractions of demand. In the same manner, we set the demand (D_{nps}) in relation to the shelf capacity (SH_{ps}) to obtain a similar fractional form and define $TSH_{nps} = D_{nps}/SH_{ps}$. Any unfulfilled demand is assumed to be lost, which leads to a penalty cost for each unit (OC_p). Also, holding costs for each unit in the store (backroom + shelves) per period (HC_p) are incurred.

3.3.3. Handling Effort and Cost Parameters

Within the warehouse, the process of picking is based on the orders of the stores. This process incurs labor efforts, such as

- a major (fixed) set-up effort both in the warehouse and store when at least one product is picked (FE^W and FE^S_s): for example the worker taking his picking cart and starting his tour,
- a minor (fixed) set-up effort both in the warehouse and store for a specific product (group) being picked (PE^W_p and PE^S_{ps}): for example the worker going to a specific bin the product is allocated to,
- a variable effort both in the warehouse and store for picking the individual items (VE^W_p and VE^S_{ps}): for example the grabbing of an item and putting it into the picking cart.

This leads to a variation in the utilization of the warehouse and store workers over the periods, which we limit to be below an upper bound of UB^W and UB_s^S in each period. We distinguish between the above parameters for manual effort (in time units) and equivalent cost parameters (in monetary units). FC^W, PC_p^W and VC_p^W are cost factors accounting for the fixed costs (major setup), product dependent fix costs (minor setup) and variable costs in the warehouse and FC_s^S, PC_{ps}^S and VC_{ps}^S are the respective cost factors in the store shelf stacking process. These cost factors are used as an input for the cost-based optimization approach of our MILP model, while the effort-based factors are used for the capacity constraint therein. Note that the cost factors may represent additional cost factors besides the handling effort, such as transportation costs.

$TVC_{ipsn} = D_{nps}\left(THC_{ipn} + VC_p^W + VC_{ps}^S\right)$ are the total variable costs of a delivery in period i completely fulfilling the demand in period n, where

$$THC_{ipn} = \begin{cases} (n-i)\, HC_p & \text{if } i \leq n \\ \left(\bar{N} - i + n\right) HC_p & \text{if } i > n \end{cases}$$

stand for the total holding costs in that case and $\bar{N}$ is the total number of periods over all sample weeks. Similarly, the total (maximum) out of stock costs per demand period are set as $TOC_{nps} = D_{nps} \cdot OC_p$.

To obtain values for the handling effort in practice, motion time studies have to be conducted as in Curşeu et al. (2008); Hübner and Schaal (2017); Sternbeck and Kuhn (2014); Van Zelst et al. (2009). These studies measure the time taken for sub-processes, such as walking, grabbing, putting, etc. over a number of e.g., picking or shelf replenishment orders. Based on the average values, the handling effort (in time units) and the corresponding costs, when weighting with the average wage of the relevant employees, can be derived.

3.3.4. Timeline

The timeline shown in Figure 3.1 consists of a base time horizon with set $T = \{1, \ldots, \bar{T}\}$, that can, for example, represent a week of six working days, each of which is divided into a morning, afternoon and evening (in this case $\bar{T} = 18$). We further define the integer $\bar{E}$ as the number of sample weeks and the total number of data periods $\bar{N} = \bar{E} \cdot \bar{T}$. The

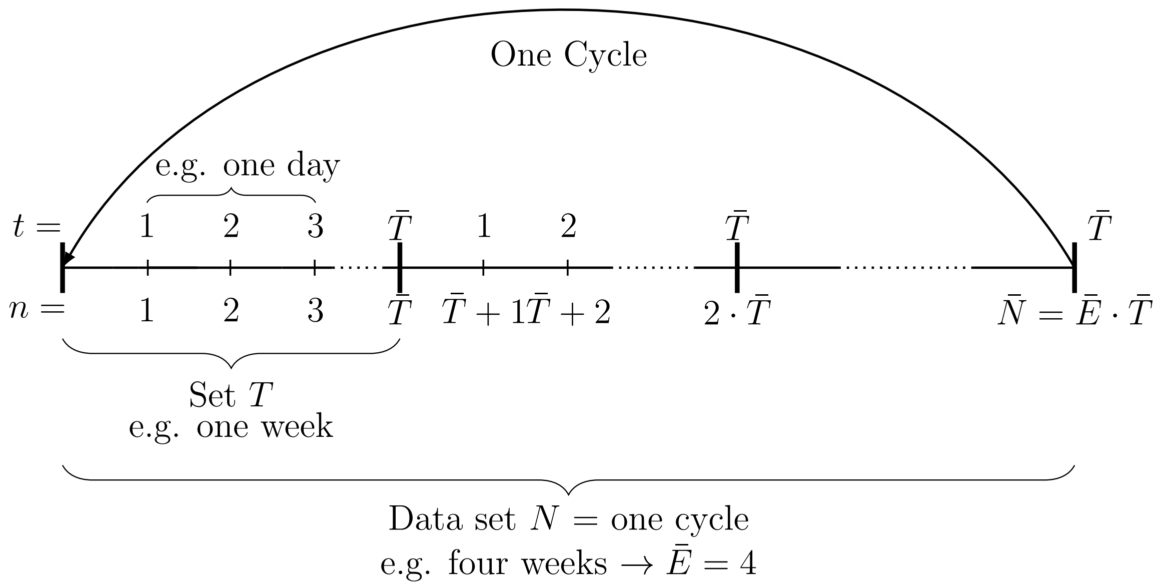


Figure 3.1.: Timeline

corresponding set of data periods is then $N = \{1, \dots, \bar{N}\}$. We define

$$\hat{t}(n) = \begin{cases} n \mod \bar{T} & \text{if } n \mod \bar{T} \neq 0 \\ \bar{T} & \text{if } n \mod \bar{T} = 0 \end{cases}$$

as the base period equivalent to data period n.

A central concept we follow is that an order is characterized by the period during which it has been placed (say period i) and by the period the demand of which the order should (partially) cover (say period n). Furthermore, we introduce $\bar{A}$ as the maximum time distance between any order and demand period. We introduce this restriction to make sure that the decisions of ordering do not reach too far into the future, as would, for instance, be the case if an order is placed today for a demand that takes place in four weeks. We allow cyclical decision making (represented by the arrow in Figure 3.1), where orders in a later period are allowed to serve earlier periods. This would indicate an order for the next cycle's demand, as the next cycle again starts with the first period. By doing so, we do not need to assume initial inventories, as is done in most of the JRP or lot-sizing literature. Initial inventories are thus set in the best possible way by our model. Although we encounter non-stationary demands within the week, this approach assumes that we solve an infinite horizon setting in its steady state. In doing so, we follow the assumptions of Zipkin (1989), who considers an infinite horizon setting where demands are non-stationary from one period to the next, but repeat themselves after a cycle.

We next define some sets of periods to be used in the mixed integer linear program. The sets are applicable to combinations of order/demand periods, as well as on combinations of shelf stacking/demand periods. For easier reading, explanations are only given for order/demand period combinations.

We define:

$$I^r(n) := \left\{ l : l \in N, \begin{array}{ll} (n - \bar{A}) \leq l \leq n & \text{if } n - \bar{A} \geq 1 \\ (l \leq n) \vee (l \geq (\bar{N} - \bar{A} + n)) & \text{if } n - \bar{A} \leq 0 \end{array} \right\} \quad \forall n \in N \quad (3.1)$$

$$N^r(l) := \left\{ n : n \in N, \begin{array}{ll} l \leq n \leq (l + \bar{A}) & \text{if } l + \bar{A} \leq \bar{N} \\ (n \leq \bar{N}) \vee (n \leq (\bar{A} + l - \bar{N})) & \text{if } l + \bar{A} > \bar{N} \end{array} \right\} \quad \forall l \in N \quad (3.2)$$

The set $I^r(n)$ includes all those periods l where it is possible to place an order covering a demand in period n. To illustrate this and the next definitions of sets, consider a setting where one period equals one day and we look at a time horizon $T = N = \{$*Monday, Tuesday, . . . , Saturday*$\}$. If the maximum number of periods between order and demand $\bar{A} = 4$, then $I^r(Wednesday) = \{$*Wednesday, Tuesday, Monday, Saturday, Friday*$\}$. In contrast, $N^r(l)$ comprises all demand periods n satisfied by an order in period l and in the example $N^r(Wednesday) = \{$*Wednesday, Thursday, Friday, Saturday, Monday*$\}$.

For the MILP formulation, we need to know which combinations of order/demand periods are to be considered for determining the inventory level in a certain period j. We thus define the set $H^+(j)$ that enumerates those combinations for a period j at the time after an order has been placed. The set H^- defined in (3.4) is just a minor modification of H^+, where orders placed at period j are not included. Hence, it represents the situation in the system before an order is placed.

$$H^{+}(j) := \{(l,n) : l \in N, n \in N^{r}(l), (l \leq j \leq n) \vee (j \leq n < l) \vee (n < l \leq j)\} \qquad \forall j \in N \quad (3.3)$$

$$H^{-}(j) := \big\{(l,n) : (l,n) \in H^{+}(j), l \neq j\big\} \qquad \forall j \in N \quad (3.4)$$

$H^{+}(j)$ includes the combinations of all those time paths that travel over j, as the products ordered along those paths are physically available in j. Here, one can distinguish between three cases: Ordering (in l) and demand (in n) occur in the same cycle ($l \leq j \leq n$), the order has been placed one cycle earlier than j ($j \leq n < l$), or the demand takes place one cycle after j ($n < l \leq j$). Applied to the earlier example, this means that $H^{+}(Wednesday)$ includes order-demand period combinations (*Monday, Thursday*), (*Tuesday, Wednesday*), ... as examples for the first case, (*Saturday, Thursday*), (*Friday, Wednesday*) for the second case and (*Wednesday, Monday*) for the third case. In $H^{-}(Wednesday)$ the combination (*Wednesday, Monday*) is not included.

Set F is used for the flow restrictions and comprises of those periods that cumulate the potential order periods relevant for a demand in period n up to the shelf stacking period l.

$$F(l,n) := \left\{ j : j \in I^{r}(n), \begin{array}{ll} j \leq l \vee j > n & \text{if } l \leq n \\ (n+1) \leq j \leq l & \text{if } l > n \end{array} \right\} \qquad \forall l \in N, n \in N^{r}(l) \quad (3.5)$$

These periods lie on the time path between one period after n and (including) period l. We again have to distinguish between two cases. The first covers those (l,n) combinations where $l \leq n$ and where orders are relevant that have been placed after n in the previous cycle or in this cycle before or during l. If $l > n$ in the second case, periods placed after n, but before or during period l are relevant, i.e. we only consider orders of the same cycle. In our example, the first case would apply for $F(Monday, Wednesday) = \{$*Thursday, Friday, Saturday, Monday*$\}$ and the second case for $F(Saturday, Wednesday) = \{Thursday,\ Friday,\ Saturday\}$.

3.3.5. Mixed Integer Linear Program (INT)

The INT model constructs delivery schedules for all product-store combinations, taking the joint replenishment effects in the warehouse and stores into account and adhering to the labor capacity constraints. In order to achieve robustness, we define that decisions on delivery patterns and order-up-to levels are valid for all sample weeks. In reference to stochastic programming literature, these decisions form the here-and-now (first-stage) decisions, for which non-anticipativity is valid, as they do not change in the sample weeks. The actual order decisions in each sample week, on the other hand, form the wait-and-see (second stage) decisions, which are bound to the first-stage decisions on the delivery pattern and order-up-to level.

As mentioned in the literature review in Section 3.2, the plant-location type formulation showed the best computational performance regarding the JRPDD in the studies of Boctor et al. (2004) and Gao et al. (2008). This formulation does not rely on a formulation of absolute ordering decisions, instead it defines the size of an order by the fraction of a future demand it can cover. This leads to an absence of the inventory balance constraint. We thus define continuous decision variables g_{ipsn} to determine the fraction of the demand of product p in period n supplied to store s in period i. h_{lpsn} is the fraction of demand of period n stacked into shelves in period l for product p at store s. To capture the out-of-stock situations, we use decision variable m_{nps} to account for the lost fraction of demand. Binary decision variables u_t^W and v_{tp}^W indicate whether a major order set-up for at least one product and a minor order setup for an individual product p takes place in the warehouse. Analogously, we introduce binary variables for a major shelf stacking set-up in store s via the variables u_{ns}^S and v_{nps}^S in order to indicate a minor shelf stacking set-up for the respective product-store combination. As introduced before, z_{tps} is the binary variable indicating whether a delivery in period t for product p and store s is allowed in the delivery schedule (when $z_{tps} = 1$) or not (when $z_{tps} = 0$). This decision has to be consistent throughout all $\bar{E}$ sample weeks. Another continuous variable oul_{tps} decides on the order-up-to level in period t for product p in store s. Note that oul_{tps} represents fractional values between 0 and 1, which can be translated to actual positive integer values for order-up-levels (OUL_{tps}) by multiplication with the sum of all demands for all periods and rounding up to the nearest integer $OUL_{tps} = \lceil oul_{tps} \sum_{n \in N} D_{nps} \rceil$.

For an overview of the notation see Table 4.1.

$$\begin{aligned}
\text{minimize} \sum_{i \in N} \sum_{p \in P} \sum_{s \in S} & \left(TOC_{ips} \cdot m_{nps} + \sum_{n \in N^r(i)} TVC_{ipsn} \cdot g_{ipsn} \right) \\
& + \sum_{t \in T} u_t^W \cdot FC^W \cdot \bar{E} + \sum_{p \in P} \left(PC_p^W \cdot \sum_{t \in T} v_{tp}^W \right) \cdot \bar{E} \\
& + \sum_{s \in S} \left(FC_s^S \cdot \sum_{n \in N} u_{ns}^S \right) + \sum_{s \in S} \sum_{p \in P} \left(PC_{ps}^S \cdot \sum_{n \in N} v_{nps}^S \right)
\end{aligned} \tag{3.6}$$

subject to

$$m_{nps} + \sum_{l \in I^r(n)} h_{lpsn} = 1 \qquad \forall n \in N; p \in P; s \in S \tag{3.7}$$

$$\sum_{j \in F(l,n)} h_{jpsn} \leq \sum_{j \in F(l,n)} g_{jpsn} \qquad \forall l \in N; p \in P; s \in S; n \in N^r(l) \tag{3.8}$$

$$z_{tps} \leq u_t^W \qquad \forall t \in T; p \in P; s \in S \tag{3.9}$$

$$z_{tps} \leq v_{tp}^W \qquad \forall t \in T; p \in P; s \in S \tag{3.10}$$

$$g_{ipsn} \leq z_{\hat{t}(i)ps} \qquad \forall i \in N; p \in P; s \in S; n \in N^r(i) | D_{nps} > 0 \tag{3.11}$$

$$v_{nps}^S \leq u_{ns}^S \qquad \forall p \in P; s \in S; n \in N \tag{3.12}$$

$$h_{lpsn} \leq v_{lps}^S \qquad \forall l \in N; p \in P; s \in S; n \in N^r(l) | D_{nps} > 0 \tag{3.13}$$

$$\sum_{(l,n) \in H^+(j)} h_{lpsn} \cdot TSH_{nps} \leq 1 \qquad \forall j \in N; p \in P; s \in S \tag{3.14}$$

$$J_{ips} = \sum_{(l,n) \in H^+(i)} g_{lpsn} \cdot D_{nps} - \sum_{(l,n) \in H^-(i)} h_{lpsn} \cdot D_{nps} \qquad \forall i \in N; p \in P; s \in S \tag{3.15}$$

$$\frac{J_{ips}}{\sum_{j \in N} D_{jps}} \geq oul_{\hat{t}(i)ps} + \left(z_{\hat{t}(i)ps} - 1 \right) \qquad \forall i \in N; p \in P; s \in S \tag{3.16}$$

Table 3.1.: Notation for model INT

Sets	
$p \in P$	Products
$s \in S$	Stores
$t \in T$	Base periods
$i, j, l, n \in N$	Data periods
$I^r(n)$	Order periods in reach for a demand in n
$N^r(i)$	Demand periods in reach for an order in i
$H^+(j)$	Order-demand period combinations, in store at the end of period j
$H^-(j)$	Order-demand period combinations, in store at the start of period j
$F(l, n)$	Order periods relevant for a demand in n and a shelf stacking in l
Parameters	
D_{nps}	Demand of product p and store s in n
FE^W, FC^W	Major setup effort and costs in the warehouse
PE_p^W, PC_p^W	Minor setup effort and costs of product p in the warehouse
VE_p^W, VC_p^W	Variable effort and costs of product p in the warehouse
FE_s^S, FC_s^S	Major setup effort and costs in store s
PE_{ps}^S, PC_{ps}^S	Minor setup effort and costs for product p in store s
VE_{ps}^S, VC_{ps}^S	Variable effort and costs for product p in store s
TVC_{ipsn}	Total variable costs of an order in i covering the demand in n for product p and store s
OC_p	Out-of-stock costs of product p per unit
TOC_{psn}	Total out-of-stock costs of the demand in n for product p and store s
HC_p	Holding costs of product p per unit and period
THC_{ipn}	Total holding costs for an order from i covering the demand in n of product p
SH_{ps}	Shelf space of product p in store s
TSH_{psn}	Total shelf space consumption of demand in n, product p and store s
UB^W, UB_s^S	Upper bound on the load in the warehouse and store s
Decision variables	
z_{tps}	Binary indicator of delivery in t, product p and store s
g_{ipsn}	Fraction of demand in n, product p and store s delivered in i
h_{lpsn}	Fraction of demand in n, product p and store s stacked on a shelf in l
oul_{tps}	Fractional order-up-to level in t, product p and store s
m_{nps}	Fraction of demand lost in n, product p and store s
u_t^W	Binary indicator of major setup in the warehouse in t
v_{tp}^W	Binary indicator of minor setup in the warehouse in t for product p
u_{ns}^S	Binary indicator of major setup in n and store s
v_{nps}^S	Binary indicator of minor setup in n, product p and store s
L_i^W, L_{ls}^S	Load in the warehouse in i and in store s in l

$$oul_{\hat{t}(i)ps} - \frac{J_{ips}}{\sum\limits_{j \in N} D_{jps}} \leq 1 - os_{ips} \qquad \forall i \in N; p \in P; s \in S \tag{3.17}$$

$$\frac{J_{ips}}{\sum\limits_{j \in N} D_{jps}} - oul_{\hat{t}(i)ps} \leq os_{ips} \qquad \forall i \in N; p \in P; s \in S \tag{3.18}$$

$$g_{ipsn} \leq 1 - os_{ips} \qquad \forall i \in N; p \in P; s \in S; n \in N^r(i) \tag{3.19}$$

$$\begin{aligned} L_i^W =& FE^W \cdot u_{\hat{t}(i)}^W + \sum_{p \in P} PE_p^W \cdot v_{\hat{t}(i)p}^W \\ &+ \sum_{p \in P} \sum_{s \in S} \sum_{n \in N^r(i)} VE_p^W \cdot g_{ipsn} \cdot D_{nps} \quad \forall i \in N \end{aligned} \tag{3.20}$$

$$L_i^W \leq UB^W \qquad \forall i \in N \tag{3.21}$$

$$\begin{aligned} L_{ls}^S =& FE_s^S \cdot u_{\hat{t}(l)s}^S + \sum_{p \in P} PE_{ps}^S \cdot v_{\hat{t}(l)ps}^S \\ &+ \sum_{p \in P} \sum_{n \in N^r(l)} VE_{ps}^S \cdot h_{lpsn} \cdot D_{nps} \qquad \forall l \in N; s \in S \end{aligned} \tag{3.22}$$

$$L_{ls}^S \leq UB_s^S \qquad \forall l \in N; s \in S \tag{3.23}$$

$$z_{tps}, u_t^W, u_{ns}^S, v_{tp}^W, v_{nps}^S, os_{nps} \in \{0, 1\} \qquad \forall t \in T; p \in P; s \in S; n \in N \tag{3.24}$$

$$oul_{tps}, g_{ipsn}, h_{lpsn} \in \mathbb{R} \qquad \forall t \in T; i, l, n \in N; p \in P; s \in S \tag{3.25}$$

The objective (3.6) comprises the total of ordering, shelf stacking, holding and out of stock costs. Major and minor set-up costs of ordering are incurred in each one of the $\bar{E}$ representations as soon as they are set by z_{tps}. Constraints (3.7) ensure that in each period the demand is either served or lost. As the fraction of demand to order g_{ipsn} and the fraction of demand to stack into shelves h_{lpsn} are represented by two different decision variables, we need the flow constraints in (3.8) to make sure that everything that is stacked in one period has to be delivered to the store in the same period or earlier. In (3.9), a major set-up in the warehouse is incurred whenever at least one product is delivered to at least one store (i.e. there is a positive fraction to order and the demand of period n it covers is positive). Likewise, it is set that minor delivery set-ups, too, can only take place in those periods achieved through constraints (3.10). The delivery of a positive fraction is identified by constraints (3.11). This constraint ensures non-anticipativity for the decision of the delivery pattern, as it forces the delivery pattern to be the same for all sample weeks. Constraints (3.12) are introduced to allow for minor set-ups in the stores only when major set-ups in the stores are scheduled and constraints (3.13) allow for positive shelf stacking fractions in those cases only. Shelf capacity constraints are placed by (3.14). In (3.15), we determine the inventory level J_{ips}. There, the set $H^{+}(j)$ indicates all orders that have been placed and are still in the store, while $H^{-}(j)$ indicates all shelf stacking events that have been placed up to the last period. Based on the subtraction of the latter set from the former, we can thus determine the current inventory level. In (3.16), we divide J_{ips} by the total demand over all periods to get a fractional expression, which is ensured to be smaller than the order-up-to level oul_{tps} only during periods where no delivery takes place (i.e. where $z_{\hat{t}(n)ps} = 0$) and bigger or equal to the order-up-to level in all other periods. The order-up-to level is then set as a value between 0 and 1, indicating the fraction of total demand over all periods ($\sum_{n \in N} d_{nps}$). Due to the non-stationarity of the setting, situations, where the inventory on-hand is greater than the order-up-to level, might arise. We deal with these cases by setting an auxiliary binary variable os_{ips} to 1 whenever said overstock occurs by applying (3.17) and (3.18). In (3.19), we define that a positive order quantity can only happen outside an overstock situation, i.e. when $os_{ips} = 0$. (3.16)–(3.18) again enforce non-anticipativity in the order-up-to levels, as the values remain the same for all sample weeks.

We restrict the load in the warehouse L_n^W in (3.21) and the load at the store level L_{ls}^S in (3.23) to values lower than their respective upper bound. L_n^W and L_{ls}^S take the total effort (major, minor and variable) of all orders of that period into account and are defined as in (3.20) and (3.22).

Finally, we define binary variables in (3.24) and continuous variables in (3.25).

3.4. Hierarchical Decomposition

Driven by the complexity of the model at hand, we generate a hierarchical decomposition approach to reduce computational effort. In the first stage, a set of possible delivery patterns is constructed as in e.g. Campbell and Savelsbergh (2004) and in the second stage, the INT model is reformulated to contain only a selection out of this set for each product store combination. By doing so, we reduce the solution space in the second stage model as, instead of all possible combinations of delivery periods, we only present the reduced set of possible combinations obtained by the first-stage model.

3.4.1. Pattern Selection

Assuming an existing set of delivery patterns $r \in R$, we can use an adjacency matrix z_{tr} and a binary decision variable o_{rps} instead of the former decision variable z_{tps}. z_{tr} defines whether or not a delivery in period t is allowed in delivery pattern r, while o_{rps} decides on whether delivery pattern r is used (when taking value 1) for product p in store s or not.

We apply the following changes with regard to the INT model:

$$\sum_{r \in R} o_{rps} \cdot z_{tr} \leq u_t^W \qquad \forall t \in T; p \in P; s \in S \tag{3.26}$$

$$\sum_{r \in R} o_{rps} \cdot z_{tr} \leq v_{tp}^W \qquad \forall t \in T; p \in P; s \in S \tag{3.27}$$

$$g_{ipsn} \leq \sum_{r \in R} z_{\hat{t}(i)r} \qquad \forall i \in N; p \in P; s \in S; n \in N^r(i) \tag{3.28}$$

$$\sum_{r \in R} o_{rps} = 1 \qquad \forall p \in P; s \in S \tag{3.29}$$

$$o_{rps}, u_t^W, u_{ns}^S, v_{tp}^W, v_{nps}^S \in \{0,1\} \qquad \forall t \in T; p \in P; s \in S; n \in N \tag{3.30}$$

where (3.26), (3.27), (3.28) and (3.30) replace (3.9), (3.10), (3.11) and (3.24). (3.29) ensures the selection of exactly one delivery pattern for each product and store.

3.4.2. Pattern Construction

We investigate three approaches for constructing the sets of delivery patterns: 1) a decomposition for each product and store combination individually (DPS), 2) a decomposition of the time horizon into sample weeks (DTW) and 3) a genetic algorithm (GAM).

3.4.2.1. Product Store Decomposition (DPS)

The product store decomposition (DPS) approach solves the model for each product and each store separately, while considering the whole time horizon N. This results in $\bar{P} \cdot \bar{S}$ sub-problems. Small changes to the original MILP of (3.6) - (3.25) are necessary. Major and minor setups defined for multiple products and/or stores are allocated equally among all sub-problems. Thus, instead of FC^W, PC_p^W and FC_s^S, the model uses $FC^W / \left(\bar{S} \cdot \bar{P}\right)$, $PC^W/\bar{S}$ and $FC_s^S/\bar{P}$. We proceed in the same way for FE^W, PE_p^W and FE_s^S. The integer decision variable indicating the delivery pattern structure z_{tps} is transformed to the decision variable z_t. Also, indices of the decision variables simplify to g_{in}, h_{ln}, m_n, oul_t, J_i, os_i, u_n^S, v_t^W and v_n^S. Furthermore, after the optimization of each sub-problem, the resulting vector z_t is added to the set of possible delivery patterns R, except if it constitutes a duplicate of an already existing delivery pattern in R.

The benefit of this approach is a vast reduction of complexity. The binary and continuous variables connecting the sub-problems of each store and product, such as u_t^W, v_{tp}^W or oul_{tps} in the original formulation, lead to a combined consideration of all product and store sub-problems for any fixed binary decision. In the DPS formulation, this is no longer the case, which is why a fixed binary decision only considers the effects in the specific product and store. On the other hand, the approach neglects the joint replenishment considerations as it optimizes schedules for products and stores separately. The pattern selection stage can only partially heal this by, for example, assigning the same schedule to two products, which might be optimal for one product but sub-optimal for another. In the original formulation, a different schedule might be derived with a better compromise for both products, which might reduce overall costs.

3.4.2.2. Time Horizon based Decomposition (DTW)

The DTW decomposes the problem by time into its sample weeks and solves the problem for all products and stores as in (3.6) - (3.25), yet with the reduced set of data periods for one sample. This decomposition of the data set N into $\bar{E}$ sub-problems leads to classical JRPDD problems. The changed subset of data periods depends on the sub-problem $k \in \{1, 2, \ldots, \bar{E}\}$ and is defined as $N^{TW}(k) := \{(k-1) \cdot \bar{T} + 1, (k-1) \cdot \bar{T} + 2, \ldots, k \cdot \bar{T}\}$. It replaces set N in the original formulation. Furthermore, the two sets $N^r(i)$ and $I^r(n)$ are replaced by the corresponding sets $N^{TWr}(i,k) := \{i, i+1, \ldots, min\{i + \bar{A}, k \cdot \bar{T}\}\}$ and $I^{TWr}(n,k) := \{max\{n - \bar{A}, (k-1) \cdot \bar{T} + 1\}, max\{n - \bar{A} + 1, (k-1) \cdot \bar{T} + 2\}, \ldots, n\}$. Note that sets $F(l,n)$, $H^+(j)$ and $H^-(j)$ are constructed as in (3.3)-(3.5), but use the sets $N^{TW}(k)$, $N^{TWr}(i,k)$ and $I^{TWr}(n,k)$ just defined.

By decomposing the problem into the time intervals as considered in the DTW approach, we reduce complexity for all those decisions that connect the sample weeks in the original formulation, e.g. z_{tps} and oul_{tps}. In contrast to the DPS, the joint replenishment effects are still considered to their full extent, i.e. across all products and stores. The shortcoming of this approach is that the resulting schedules are not guaranteed to be robust in the other time intervals. This is again healed to some extent in the second stage by choosing the most robust schedules from the set.

3.4.3. Genetic Algorithm (GAM)

We implement GAM in the GA toolbox environment of Matlab R2014b and use genetic operators as well as scaling and selection approaches thereof.

3.4.3.1. Representation and Initialization

The chromosome in GAM represents the binary delivery schedule z_{tps}. This leads to the output of one (best) schedule being used for the second stage model of Section 3.4.1. For a starting population, we use the outcome of the savings heuristic of Section 3.5.3 for one individual solution and several randomly generated solutions of the remaining population.

3.4.3.2. Evaluation

The value of each chromosome is obtained via a computation of the total costs over the considered data periods based on the objective in (3.6). Within GAM, we approximate order-up-to levels via finding the smallest integer value (positive or zero) for $O\bar{U}L_{tps}$ that satisfies (3.31) in a bisection search. (3.31) is based on the findings of Chiang (2006) for optimal base stock levels of (R, S) policies in a stationary demand, lost sales and zero lead time setting. For an optimal value for $O\bar{U}L_{tps}$, he finds that

$$(HC_p + OC_p) \cdot \left(\sum_{i=1}^{|f(t)|} \Phi_i \left(O\bar{U}L_{tps} \right) \right) - OC_p \left(\sum_{i=1}^{|f(t)|-1} \Phi_i \left(O\bar{U}L_{tps} \right) \right) - \\ \left(VC_p^W + VC_{ps}^S \right) \cdot \Phi_{|f(t)|} \left(O\bar{U}L_{tps} \right) \geq |f(t)| \cdot HC_p \tag{3.31}$$

where $f(t)$ represents the set of all periods including t up to one period before the next delivery is scheduled and $|f(t)|$ is the cardinality of the set. Again, the cyclical approach allows for $f(t)$ to include periods with an index smaller than t. $\Phi_i(a) = Pr(x_i > a)$ represents the complement of the combined cumulative historical distribution function of x_i, the sum of demand of the first i periods in $f(t)$. Note that we directly estimate $O\bar{U}L_{tps}$, which takes a positive integer value, rather than oul_{tps} of the MILP formulation introduced earlier, which is limited to real values between 0 and 1.

Furthermore, whenever a product is out-of-stock on the shelf and inventory is available in the backroom, shelf stacking is initiated, which means that the shelf space of the product will be filled up. Although shelf stacking quantities might be fewer than necessary for filling up the shelf in the MILP, it is always beneficial to do so because of major setups in the shelf stacking process.

As, in GAs in general, hard constraints might lead to a premature convergence of sub-optimal solutions (see, e.g., Michalewicz & Schmidt, 2007), we set a penalty for the relative (positive) violation of the upper bound capacity constraints in both the warehouse and the store that is equal to the respective fixed cost value (i.e. FC^W resp. FC_s^S). Thus, we add the terms

$$\sum_{i \in N} FC^W \cdot \left(\frac{L_n^W}{UB^W} - 1 \right)^+ + \sum_{l \in N} \sum_{s \in S} FC_s^S \cdot \left(\frac{L_{ls}^S}{UB_s^S} - 1 \right)^+ \tag{3.32}$$

to the objective function, where $(a)^+ = max\{a, 0\}$.

3.4.3.3. Strategies and Operators

Scaling Strategies To scale the raw values obtained by the evaluation, the approaches of *Ranking*, *Top*, *Proportional* or *Shift Linear* can be applied. When *Ranking* is applied, the values of the individuals, rather than an actual score, are associated with their rank (1,2, etc.), which places each individual in equal distance to the next. The *Top* approach selects a predefined percentage (the scaling parameter) of the best individuals in a population and assigns them with an equal probability value and the remaining individuals with a value of zero. *Proportional* scaling sets the scaled value proportionally to the raw value of the individuals. Lastly, the *shift linear* approach sets the fittest individual to a survival rate defined upfront multiplied by the average score over all individuals and scales all other individuals accordingly.

Selection of Parents The strategies for the selection of parents are *Uniform*, *Roulette*, *Remainder* and *Tournament*. In the *Uniform* selection strategy, parents are chosen randomly based on a uniform distribution set by the scaled values of the individuals and the number of parents. The *Roulette* approach acts out a game of roulette, where each individual is associated with a segment the size of which is in accordance with its scaled value. Then, the segments are randomly chosen as parents with a probability that is equal to its size segment. A mixed approach is performed by the *Remainder* strategy, where the integer part of the scaled value of the individuals leads to a deterministic selection and the fractional part of the scaled value is entered into the *Roulette* approach for a random selection. In a *Tournament*, a specific number of individuals is chosen at random and the best of those individuals, based on their scaled values, is selected as parent.

Genetic Operators Within the GA, we create children for the next generation, where the crossover fraction defines the number of offspring to be produced by a crossover operation and the rest is produced via a mutation operation. The mutation strategy is based on a selection of genes with a uniform probability, where each selected gene in the chromosome is changed individually (i.e. to 1 if being 0 or vice versa).

For the crossover strategies, *Scattered*, *Single Point* and *Two Point* are investigated. A random binary vector in the size of the chromosome is first created in the *Scattered* approach where a 1 indicates that the gene of the first parent is chosen and a 0 indicates that the gene of the second parent is chosen for the child. The *Single Point* approach takes the first n genes, with n being a randomly generated number, from the first parent and the rest from the second parent. An extension of this is the *Two Point* approach, where another random number m indicates up to which point the genes of the second parent (genes $n + 1$ to m) are chosen. After the mth gene, the genes of the first parent are again used.

3.4.3.4. Metacalibration of GAM

GAM uses methods of the Matlab GA toolbox for scaling the fitness value, selecting the parents and applying genetic operators. As in each case the possibilities of chosen strategies, along with their parameter values, are manifold, we develop a model to select appropriate settings for any instance. We develop a self-adaptive genetic algorithm (SGA) that searches for the best settings of a given instance. Within the SGA, the evaluation of the fitness function is the outcome of the GAM approach with a specific set of input parameters. Population size, elite fraction, fitness scaling strategy and parameter, parent selection strategy and parameter, crossover strategy and parameter, as well as mutation fraction, are part of the search. These parameters are translated into an array where strategies are represented by integer values and fractions and parameters are represented as real values each of which has a specified upper and lower bound. This array represents the chromosome in the SGA. When running the SGA, we thus ultimately receive the best suitable parameter settings for a certain instance.

As in a practical setting we do not want to rerun the SGA for every new instance, we develop the following approach: We apply the SGA on multiple calibration instances and fit regression models to the results, which can then be used for deriving settings for all other instances. The motivation behind this approach is to perform a metacalibration in which parameters and strategies are set by an analysis of best settings fitted to calibration instances. The general idea of metacalibration can also be found in Vidal et al. (2012). However, we acknowledge that, depending on the problem instances, different parameter settings might be favorable, which is why, in our approach, we predict those settings based on regression models.

After running the SGA over all calibration instances and obtaining the best parameter settings for those instances, linear regression models for all real values (i.e. strategy parameters) and multinomial regression models for all categorical settings (i.e. chosen strategies) are created. Based on the regression models, strategies and parameters can thus be estimated for the instances of the numerical study without having to tune the parameters individually for every instance, as this would imply the repeated running of the model, which would not allow a fair comparison with the other models in terms of maximum solution time. Thus, managers who use the model do not have to possess the knowledge on how to tune the parameters of a GA, but can instead use the metacalibration for an automated setting of parameters.

3.5. Benchmark Models

3.5.1. Sequential Model (SEQ)

We sequentially optimize two decisions of the INT model in the SEQ model. Initial decisions are made on the delivery patterns and order-up-to levels, thus it is assumed that no shelf stacking decision has to be made. The model uses the first two lines of objective (3.6), i.e. all warehouse and order relevant costs plus the constraints in (3.7), (3.9) - (3.11), (3.15) - (3.21) and the relevant constraints of (3.24) and (3.25). Note that in (3.7) variable h_{lpsn} is replaced by g_{ipsn} and in (3.15) the last sum on the right hand side is deleted.

After running this model, another optimization is applied to the shelf stacking decisions in a second stage model. This second stage model takes the solution on delivery periods and quantities as a given input and thus only optimizes on the remaining shelf stacking decisions. It uses the full objective of (3.6) and constraints (3.7) - (3.8), (3.12) - (3.14), (3.22) - (3.23) and again the relevant constraints of (3.24) and (3.25). Note that g_{ipsn}, oul_{tps}, z_{tps}, v_{tp}^{W} and u_{t}^{W} are given by the first stage and thus enter the model as input parameters.

By this approach, we can determine the value of coordination when taking both decisions into account simultaneously, as opposed to a decoupled decision making.

3.5.2. Expected Demand Model (EDM)

Another benchmark model EDM is introduced, where we explicitly do not consider the data-driven approach as presented in Section 3.3.5. Instead, we use the average demand over all sample weeks to create an average demand value for each base period, i.e, $\bar{D}_{tps} = 1/\bar{E} \cdot \sum_{k=1}^{\bar{E}} D_{\hat{n}(t,k)ps}$, where $\hat{n}(t,k) = t + (k-1) \cdot \bar{E}$. We introduce the following constraints into the model formulation:

$$J_{ips} \geq \gamma \cdot \bar{D}_{ips} \quad \forall i \in T; p \in P; s \in S \tag{3.33}$$

$$L_i^W = FE^W \cdot u_i^W + \sum_{p \in P} PE_p^W \cdot v_{ip}^W + \sum_{p \in P} \sum_{s \in S} VE_p^W \cdot oul_{ips} \cdot \sum_{j \in T} \bar{D}_{jps} \quad \forall i \in T \tag{3.34}$$

$$G_{lps} \geq VE_{ps}^S \cdot SH_{ps} \cdot H_{lps} \quad \forall l \in T; p \in P; s \in S \tag{3.35}$$

$$G_{lps} \geq (oul_{lps} - H_{lps}) \cdot VE_{ps}^S \cdot \sum_{j \in T} \bar{D}_{jps} \quad \forall l \in T; p \in P; s \in S \tag{3.36}$$

$$L_{ls}^S = FE_s^S \cdot u_{ls}^S + \sum_{p \in P} PE_{ps}^S \cdot v_{lps}^S + \sum_{p \in P} G_{lps} \quad \forall l \in N; s \in S \tag{3.37}$$

$$H_{lps} \in \{0,1\} \quad \forall l \in T; p \in P; s \in S \tag{3.38}$$

(3.34) and (3.37) replace (3.20) and (3.22). (3.33), (3.35), (3.36) and (3.38) are added to the original model formulation. (3.33) ensures that the inventory level in each period is at least as high as a predefined γ multiple of the average period's demand. γ thus represents the periods of supply. In doing this, we realize that a safety stock might be taken into account. (3.34) restricts the utilization in the warehouse by assuming that the whole order-up-to level oul_{ips} has to be ordered in maximum. This constraint thus ensures that the capacity constraint also holds in a worst case scenario. The constraint on the store capacity is adapted by (3.35)–(3.38). For this purpose, the binary variable H_{lps} and the continuous variable G_{lps} are introduced. G_{lps} takes the minimum value of the effort of either replenishing the whole shelf space via constraint (3.35) or the whole order-up-to level by constraint (3.36). H_{lps} indicates whether the minimum is given by the shelf space (if $H_{lps} = 1$) or the order-up-to level (if $H_{lps} = 0$). G_{lps} replaces the formulation of the variable effort in the store capacity constraint, leading to (3.37). (3.38) defines H_{lps} as a binary variable.

As we get deterministic demand for all periods in the base time horizon, the model represents the classical JRPDD approach. To find the best value for γ in each instance, we backtest the cost of the inventory system by simulating over the in-sample data, which is the same as for the data-driven approaches, and select the cost-minimal γ setup.

3.5.3. Savings Heuristic (SAVS)

Up to this point, all models are generic and applicable to many problem settings. In a last benchmarking approach, we introduce a savings heuristic (SAVS) based on the algorithm in Minner (2009). We extend it to the cyclical approach, also taking the minor setup cost of ordering into account and then we apply it to our non-stationary stochastic demand setting.

We first start with the initial lot-for-lot solution that assumes every base period t is ordering all products. For the determination of the quantities to be ordered, we assume that we order for all stores together. The order quantity in each period is equal to the order-up-to level OUL_{tps} as determined in (3.31), both here and in the following steps. For all (t, p) where $\sum_{s \in S} OUL_{tps} > 0$, we set the binary indicator $v_{tp} := 1$. Capacity slack values are computed for each period, based on the major, minor and variable effort consumed by the ordered products and their order quantities. Furthermore, for each period, $M(t)$ denotes the next major order period after t (i.e. the next period where an order of any product takes place) and $m(p, t)$ denotes the next minor order period of product p after t. (Note that this next order period, in both cases of $M(t)$ and $m(p, t)$, might have a value smaller than t due to the cyclical approach). Next, major savings ($MS(t)$) through combining the orders of all products and minor savings ($ms(p, t)$) through combining the order of one specific product of the next order period with the current order period t are calculated as

$$MS(t) := FC^W + \sum_{p \in P} PC_p^W \cdot v_{tp} \cdot v_{M(t)p} - \sum_{p \in P} \sum_{s \in S} (O\hat{U}L_{tps} - OUL_{tps}) \cdot hd_{tM(t)} \quad (3.39)$$

$$ms(p, t) := PC_p^W \cdot v_{tp} \cdot v_{m(p,t)p} - \sum_{s \in S} (O\hat{U}L_{tps} - OUL_{tps}) \cdot hd_{tm(p,t)} \quad (3.40)$$

where $O\hat{U}L_{tps}$ is again calculated based on (3.31), assuming that the next replenishment period is the one after $M(t)$ or after $m(tp)$ and

$$hd_{ij} := \begin{cases} (j-i)\, HC_p & \text{if } i \leq j \\ \left(\bar{T} - i + j\right) HC_p & \text{if } i > j \end{cases}$$

determines the cumulated holding cost factor between periods i and j. Resulting positive savings are sorted in descending order and iteratively processed. As soon as the biggest saving is identified, feasibility is checked. The amount of the additional capacity needed in t is determined and it is also tested whether the available slack capacity is enough to cope with the change. The needed capacity for a major saving (CP_t) and a minor saving (cp_{tp}) is calculated via

$$CP_t := \sum_{p \in P} (1 - v_{tp}) \cdot v_{M(t)p} \cdot PE_p^W + \sum_{s \in S} (O\hat{U}L_{tps} - OUL_{tps}) \cdot VE_p^W \qquad (3.41)$$

$$cp_{tp} := (1 - v_{tp}) \cdot v_{m(t,p)p} \cdot PE_p^W + \sum_{s \in S} (O\hat{U}L_{tps} - OUL_{tps}) \cdot VE_p^W. \qquad (3.42)$$

If infeasible and thus not realized, the respective targeted saving ($MS(t)$ or $ms(p,t)$) is set to zero. If a major saving is realized, the values for the major saving and for all minor savings of the order period $M(t)$ now included in the order of period t are set to zero (i.e. $MS(M(t)) := 0$ and $ms(p, M(t)) := 0 \quad \forall p \in P$). The same is done for the binary indicators (i.e. $v_{M(t)p} := 0 \quad \forall p \in P$). Afterwards, the next major order period ($M(t) := M(M(t))$) and all subsequent minor order periods ($m(p,t) := m(p, M(t)) \quad \forall p \in P$) are updated accordingly. The minor and major savings in period t are then updated based on (3.39). If a minor saving is realized, the same is done for the targeted product, its next minor order period and the relevant minor savings values. The algorithm continues by sorting and selecting the biggest positive saving until no more positive saving can be found.

3.6. Numerical Study

We investigate the different models with respect to their applicability to a variety of problem setups. By comparing them with the EDM model, we answer the question: how much can be gained by including the data-driven approach into the decision making process? Furthermore, we investigate the impact of problem sizes on the solution quality of the different approaches and derive guidance on which one to select, especially in large-scale instances.

3.6.1. Data Instances

We first set up a basic configuration for 10 products. Mean demand μ_p is uniformly random on $[0, 100]$, the mean coefficient of variation CV_p uniformly random on $[0.1, 0.5]$, a lower variance $var_p^L = (0.5 \cdot CV_p \cdot \mu_p)^2$ and a higher variance $var_p^H = (1.5 \cdot CV_p \cdot \mu_p)^2$. Furthermore, we draw the one period holding costs of each product uniformly random from $[0.01, 0.1]$.

To generate demand data for an instance that covers a specified number of sample weeks $\bar{E}$, we adapt and extend the approach of Kirca (1995). In addition to the mean and variance values introduced above, we have another parameter indicating the ratio of mean demand in each sample against the long-term mean. This value $\kappa_{\hat{e}(n)p}$ varies between products and sample weeks, where $\hat{e}(n) = \lceil \frac{n}{T} \rceil$ returns the sample belonging to data period n. Based on the sampling approach specified per instance, we sample the values for κ_{ep}, either using Monte-Carlo or descriptive sampling. To get κ_{ep}, we draw a value from the interval $[0.5, 1.5]$ uniformly random in the Monte-Carlo sampling approach. In the descriptive sampling approach, we randomly select equally spaced samples (see Saliby, 1990), but only let every sample occur once. Note that any other sampling method could be applied, for example, one that focuses on capturing seasonality switches or some simple ruling on causal factors such as weather. Additionally, we extract the average retail fractions of each base period t in the weekly demand as rd_t from data of the European retail chain to capture the seasonality within the week.

Generating demand, we draw two uniformly distributed random values ($\bar{Z}_{nps}^1$ and $\bar{Z}_{nps}^2$)

from $[0,1]$ for each data period, product and store. We set the variance to

$$v\bar{a}r_{nps}\left(\bar{Z}^2_{nps}\right) = \begin{cases} var^L_p & \text{if } \bar{Z}^2_{nps} > 0.5 \\ var^H_p & \text{if } \bar{Z}^2_{nps} \leq 0.5 \end{cases}$$

and the mean demand of that period to $\bar{\mu}_{np} = \kappa_{\hat{e}(n)p} \cdot rd_{\hat{t}(n)} \cdot \mu_p$. We then draw the demand values as in

$$D_{nps} = \begin{cases} \left\lceil \left(F^{-1}_{NORM}(\bar{Z}^1_{nps}; \bar{\mu}_{np}; \sqrt{v\bar{a}r_{nps}})\right)^+ \right\rceil & \text{if } v\bar{a}r_{nps} \leq \bar{\mu}_{np} \\ F^{-1}_{NB}(\bar{Z}^1_{nps}; \hat{p}; \hat{r}) & \text{if } v\bar{a}r_{nps} > \bar{\mu}_{np} \end{cases}, \qquad (3.43)$$

where

$$\bar{\mu}_{np} = \frac{\hat{r}(1-\hat{p})}{\hat{p}},\ v\bar{a}r_{nps} = \frac{\hat{r}(1-\hat{p})}{\hat{p}^2} \text{ and } 0 \leq \hat{p} \leq 1$$

Whenever $v\bar{a}r_{nps} \leq \bar{\mu}_{np}$, we use the normal cumulative distribution function (c.d.f.) F_{NORM}, which is truncated for values below zero. On the other hand, we use the negative binomial c.d.f. F_{NB} when $v\bar{a}r_{nps} > \bar{\mu}_{np}$. As has been shown by Agrawal and Smith (1996), the negative binomial distribution is a good fit for retail demand.

Next we determine values for the major and minor setups. We define parameter TBO as the time between orders, parameter MSR as the ratio between major and minor setup cost and parameter ORR as the ratio between order and shelf stacking costs. Using these, along with the average demand

$$\bar{D} = \frac{\sum\limits_{n\in N}\sum\limits_{p\in P}\sum\limits_{s\in S} D_{nps}}{\bar{N}\cdot\bar{P}\cdot\bar{S}}$$

and the average holding cost $\bar{HC} = 1/\bar{P} \cdot \sum_{p\in P} HC_p$, we first determine the average total fix cost

$$T\bar{F}C = \left(TBO^2 \cdot \bar{D} \cdot \bar{HC}\right)/2$$

by the EOQ formula. We then draw the major and minor cost factors from a normal distribution $N(\mu, \sigma)$ with

$$
\begin{aligned}
FC^W &\sim N\left(\bar{FC}^W, 0.1 \cdot \bar{FC}\right), \text{ where } \bar{FC}^W = MSR \cdot ORR \cdot T\bar{FC} \cdot \bar{P} \cdot \bar{S}, \\
PC_p^W &\sim N\left(\bar{PC}^W, 0.1 \cdot \bar{PC}^W\right), \text{ where } \bar{PC}^W = (1 - MSR) \cdot ORR \cdot T\bar{FC} \cdot \bar{P}, \\
FC_s^S &\sim N\left(\bar{FC}^S, 0.1 \cdot \bar{FC}^S\right), \text{ where } \bar{FC}^S = MSR \cdot (1 - ORR) \cdot T\bar{FC} \cdot \bar{P} \text{ and} \\
PC_{ps}^S &\sim N\left(\bar{PC}^S, 0.1 \cdot \bar{PC}^S\right), \text{ where } \bar{PC}^S = (1 - MSR) \cdot (1 - ORR) \cdot T\bar{FC}.
\end{aligned}
$$

Parameters VC^W, VC^S are set to $10 \cdot C_h$ and we assume that the effort parameters (FE, PE, etc.) take the same values as their cost parameter counterparts. Another parameter specified for each instance indicates the ratio of out-of-stock (OOS) to average total ordering and replenishment cost ($OOS - ratio$), which allows us to compute the out-of-stock cost values for each product via

$$
OC_p = (OOS - ratio) \cdot \left(TBO \cdot HC_p + VC_p^W + \frac{\sum_{s \in S} VC_{ps}^S}{\bar{S}} \right).
$$

We assume that the shelf space of a specific product in a specific store is allocated non-restrictively by taking the demand point over all demand periods that covers at least 99 % of the cumulative demand distribution of the samples drawn multiplied by the TBO. Warehouse and store capacities are chosen based on the fixed effort needed to order anything at all plus an instance specific capacity factor β as a percentage of the capacity needed on average during a given delivery period, assuming that all products (for all stores) are ordered in this period with a quantity that covers the average demand over the TBO horizon:

$$
UB^W := FE^W + \beta \cdot \left(\sum_{p \in P} PE_p^W + \sum_{p \in P} VE_p^W \cdot \bar{D}_{ps} \cdot TBO \right) \tag{3.44}
$$

$$
UB_s^S := FE_s^S + \beta \left(\sum_{p \in P} PE_{ps}^S + \sum_{p \in P} VE_{ps}^S \cdot \bar{D}_{ps} \cdot TBO \right) \quad \forall s \in S \tag{3.45}
$$

It should be noted that, for the data generation, it is assumed that all parameters generated are independent of each other.

To define our numerical design space, we first fix the time horizon $\bar{T}$ to 18 for all instances, accounting for six working days with each having a morning, afternoon and evening period. Each instance is furthermore characterized by the number of Products $\bar{P}$, the number of stores $\bar{S}$, the number of sample weeks $\bar{E}$, the cost influencing parameters TBO, MSR, ORR and OOS-ratio, the capacity level β, as well as an indication to use Monte-Carlo or descriptive sampling. These 9 characteristics span a multi-dimensional design space where each dimension is bounded as indicated in Table 3.2.

Table 3.2.: Numerical design space

Factor	Lower bound	Upper Bound	Values
Products ($\bar{P}$)	2	10	Integer
Stores ($\bar{S}$)	2	10	Integer
Capacity (β)	0.5	1.5	Real
TBO	2	10	Integer
MSR	0.1	0.9	Real
ORR	0.1	0.9	Real
OOS-ratio	1	20	Real
No. sample weeks ($\bar{E}$)	5	15	Integer
Sampling method	Monte-Carlo	Descriptive	Categorical

To obtain a good coverage of the 9-dimensional design space, we create quasi-random design points based on a Sobol sequence. A Sobol sequence creates quasi-random numbers along a multidimensional space while filling out the space and avoiding values at the boundaries of the dimensions (Santner et al., 2003). For a similar approach towards experimental design in a high-dimensional setting, see Löhndorf and Minner (2013) and for a thorough review on the matter of the appropriate design of numerical experiments, see V. C. P. Chen et al. (2006).

Lastly, we vary the value of $\gamma = 0, 1, 3, 6$ to be used in model EDM, to observe the effects of the periods of supply considered.

3.6.2. Results

We create 200 instances from the Sobol sequence and conduct our experiments on a 64-bit system with an Intel Core i7 core with 3.6 GHz and 32 GB of RAM. The created dataset along with all models can be found in an online repository under http://www.logprojects.wi.tum.de/publications/. The mixed integer programming models are solved in Xpress (Version 7.6) and the genetic algorithm, as well as simulation models, are realized in Matlab R2014b. We execute all models until a maximum time interval has elapsed or up to the solver's optimality gap $\left(= \frac{\text{objective value} - \text{lower bound}}{\text{objective value}}\right)$ of 0.01 % if at least one feasible solution has been found in the last model stage. The maximum time is set to $1,000$ seconds on every stage of the two-stage models and to $2,000$ seconds for the EDM as it only has one stage. We proceed like this in order to gain insights into the scalability of the models. After terminating, most of the approaches will remain with a positive gap, revealing which ones are best capable of closing the gap, at the same time providing better solutions.

For the meta-calibration of GAM, we randomly create 50 calibration instances in addition to the numerical study and let the SGA of Section 3.4.3.4 find best strategies and parameters for every calibration instance. We run the SGA with a population of 10, an elite count of 5, and a termination criterion of $1,000$ seconds of computation time. Apart from the parameters determined by the chromosome of the SGA, we set a termination criterion of 100 seconds for the GAM in the evaluation. As the output of the SGA, we obtain the best settings for a given instance. Using this set of parameters, we develop the regression models as outlined in Section 3.4.3.4 and use those models to predict the best parameter settings for our numerical study. To get feasible values for bounded variables on $]0,1[$, prediction values are truncated below 0.01 and above 0.99, respectively.

We compare the solutions obtained by simulating the average weekly cost of the inventory system, as in the fitness function of the GAM approach, by calculating (3.6) and adding 200 % overtime costs of the original cost factors. In order to do so, we create 200 sample paths with 26 weeks of demand each. In every simulation of a given sample path, the warm-up period is determined by the Marginal Standard Error Rule (MSER-2) approach of White (1997) (see also Law, 2015) and solutions are truncated accordingly. In order to ensure the significance of the obtained results, we set the number of sample paths to consider in each instance i (SP_i) in such a way that, ideally, they form a 95 % confidence interval (based on the standard normal distribution as all replication sizes are > 31) within a tolerance of $\epsilon = 1$ % (Law, 2015), with a maximum of 200 sample paths per instance. As a performance indicator, we use the mean relative deviation from the cost minimal policy (ζ_{ij}) for model j in instance i with

$$\zeta_{ij} = 1/SP_i \cdot \sum_{r=1}^{SP_i} \left(\left(Cost_{rij} - \min_{k \in M} Cost_{rik} \right) / \min_{k \in M} Cost_{rik} \right),$$

where $Cost_{rij}$ are the average weekly costs of run r in instance i under model j and M is the set of all models in the analysis.

As a result, we get the values for the average solution quality of all models as depicted in Table 3.3. We can see that GAM achieves the most robust solution quality over all instances. Further insights into this and the other models' performances, along with an explanation for what lies behind, are investigated in the subsequent Sections.

Table 3.3.: Average ζ_{ij} over all instances

DTW	DPS	GAM	SEQ	SAVS	EDM
32.42 %	37.87 %	**7.18 %**	27.93 %	35.67 %	10.20 %

In bold: Minimum average ζ_{ij} over all models

3.6.2.1. Significant Influences on Solution Quality

Next, we identify drivers for the performance of the models. Table 3.4 depicts the result of an Analysis of Variance (ANOVA) test conducted on a linear regression for each model answering the question how well the results are explained by the factors characterizing the instances. The purpose of this analysis is to better understand in which problem settings the models perform better or worse in order to be able to give a recommendation which model to use. The estimated coefficients β_x and the p-values of each factor in each model are depicted. Bold values highlight p-values ≤ 0.01, indicating a significant impact. As a bigger ζ_{ij} value indicates a poor model performance, a positive estimate of the coefficient refers to a worsening performance.

Since it provides, on average, the best solution over all instances, we especially compare all models with the GAM. Starting with the DTW model, where the first significant factor is the capacity having a negative coefficient, we can conclude that, in instances with loose constraints, the approach works better than the other approaches. If constraints are not tight, there are more possibilities for the optimization and computational

Table 3.4.: p-Values for all factors in all models with regard to ζ_{ij}

Model	DTW		DPS		GAM		SEQ		SAVS		EDM	
Factors	β_x	p	β_x	p	β_x	p	β_x	p	β_x	p	β_x	p
Products	0.00	0.97	-0.01	0.73	0.00	0.48	0.02	**0.00**	0.00	0.81	0.01	**0.01**
Stores	0.02	0.17	-0.02	0.30	0.00	0.17	0.01	0.03	0.01	0.30	0.01	**0.00**
Capacity	-1.03	**0.00**	-0.54	**0.00**	-0.01	0.82	0.15	**0.00**	0.38	**0.00**	0.00	0.46
TBO	-0.03	0.02	0.01	0.73	-0.01	**0.01**	0.02	**0.00**	0.05	**0.00**	0.00	0.05
MSR	-0.03	0.81	-0.22	0.22	0.00	0.91	-0.15	**0.01**	-0.04	0.53	0.00	0.46
ORR	-0.08	0.60	-0.21	0.25	0.00	0.89	-0.12	0.05	0.13	0.05	-0.04	0.08
OOS-ratio	0.03	**0.00**	0.02	**0.00**	0.00	0.54	0.01	**0.00**	0.03	**0.00**	0.01	**0.00**
No. sample weeks	0.03	**0.00**	0.01	0.35	-0.01	**0.00**	-0.01	0.06	-0.01	0.12	0.00	0.75
Sampling meth.	0.15	0.03	-0.04	0.63	-0.01	0.55	-0.05	0.08	0.00	0.93	0.01	0.27

In bold: p-Values ≤ 0.01

complexity is lower, leading to smaller gaps and thus smaller differences between the approaches. The OOS-ratio with a positive coefficient leads to the following insight: The base stock levels in the DTW are smaller (on average by 5.84 %) than GAM, which has the effect that the more important the OOS penalty, the worse the performance in comparison to the GAM as out-of-stock situations are more likely to occur. The positive coefficient for the number of samples weeks indicates that performance worsens when the number of sample weeks rises. The reason behind lies in the comparison with other approaches that use the full number of sample weeks during pattern creation. Those approaches can benefit from the more accurate depiction of the underlying unknown distribution, achieving more robust solutions, whereas the DTW cannot as it only takes one sample week into account.

For the DPS, the effects of both, the capacity and the OOS-ratio, are comparable to the effects in DTW. Also, base stock levels of DPS are on average 7.26 % smaller than GAM.

Significant effects on the GAM approach can be identified by TBO and the number of sample weeks, both with negative coefficients. The approach clearly benefits from the ease of calculation in the evaluation function. Having a bigger TBO makes solutions more complex but has no significant impact on calculation time, as the diversification procedures are not influenced by the TBO value. A larger number of sample weeks captures the demand structure better and thus brings an advantage against the EDM, which does not benefit from this, as we can see in Figure 3.2. There, one can observe that with a larger number of sample weeks used, GAM will perform better than EDM. Testing the influence of the number of sample weeks on the difference between both models in a one-factor linear regression, i.e., $(\zeta_{i,EDM} - \zeta_{i,GAM}) \sim$ No. Samples, shows a significant linear relation (p-value of 0.0169) and a positive coefficient. Thus, with a rising number of sample weeks, the GAM approach performs better than the EDM. This gap hence provides insights into the value of incorporating the data-driven approach into GAM. Another interesting insight gained from Figure 3.2 is the reduction of the variance in the solution quality for GAM with an increasing number of sample weeks. As opposed to this, EDM has a larger variance in its performance, which does not change with an increased number of sample weeks. Hence, this figure also underlines the robustness of the data-driven approach.

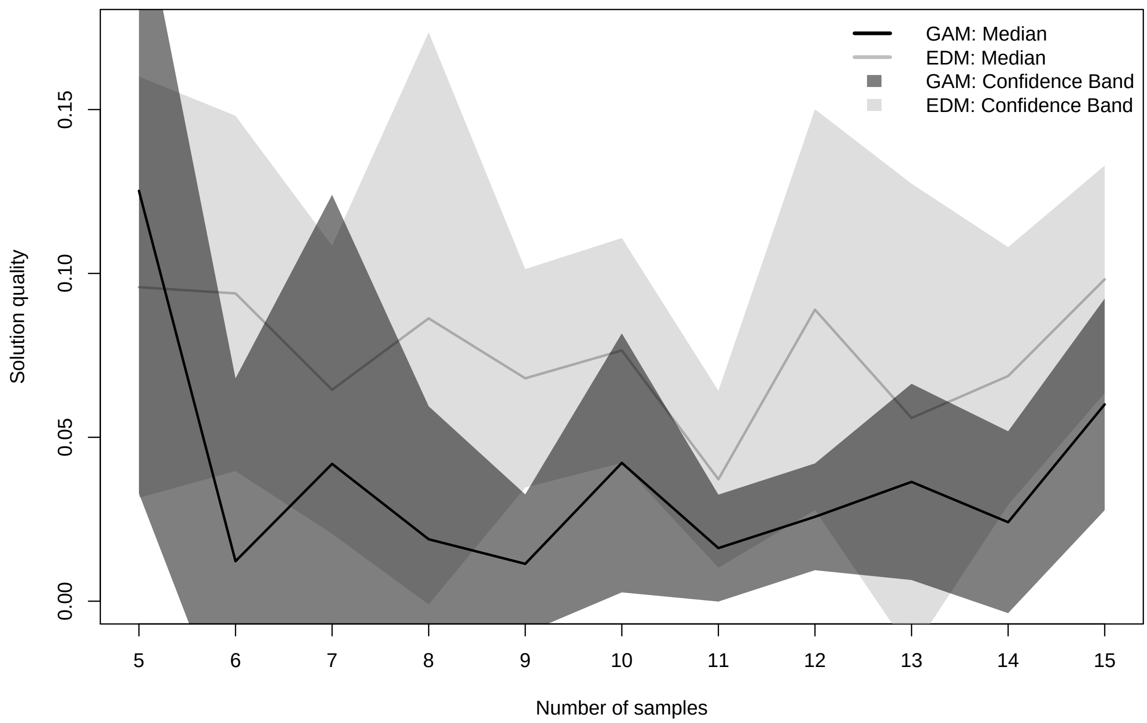


Figure 3.2.: Solution quality ζ_{ij} of GAM and EDM in reference to the number of sample weeks

In the SEQ model, the parameters products, capacity, TBO, and OOS-ratio, with positive, as well as MSR, with a negative coefficient, are significant. The significant impact of the number of products comes from the joint replenishment effect of the shelf stacking decisions. This effect is neglected during order decisions in SEQ, yet its importance increases with more products to consider. The effect of capacity and TBO lies within the tendency to order more and less often when these values rise. The model ignores shelf capacities at the ordering stage, which leads to more overstocking than other models, also in cases where demand cannot be fulfilled due to the restricted shelf space. The influence of MSR follows the reasoning as before. Due to ignoring shelf capacities, the model decides more often towards achieving joint replenishments than the other models. Thus, when MSR is high, savings in major setups have a bigger influence on the costs. In comparison with GAM, the SEQ on average has a 28.33 % larger total base stock level.

The savings heuristic SAVS performs worse when TBO, capacity or the OOS-ratio increase. This model, in particular, is driven by capacity and a higher TBO leads to higher fixed efforts in the order periods, through which the slack capacity diminishes. Therefore, it is harder for the heuristic to combine order periods, as it will only do so for the whole order quantity or not at all. The worsening effect of a higher capacity can only be explained in context with the other models. As capacity increases, the computational complexity of the more elaborated model decreases and solutions with smaller gaps are found. However, this does not affect the calculations in the simple savings heuristic, i.e. the heuristic is not benefiting as much. An explanation for the effect of the OOS-ratio lies in the approximative setting of the order-up-to level in (3.31). This shows that other models are better suited to setting the order-up-to level when more information is available. On average, the total base stock level is the lowest, with 22.72 % less than GAM.

The performance of EDM worsens significantly with a higher number of products or stores, and bigger values for the OOS-ratio. Regarding the influence of the OOS-ratio, we can reason that EDM is not able to adjust the order-up-to levels to the individual demand peaks under the capacity constraint well enough. This leads to lost sales in those demand peaks, whose impact increases with higher values for the OOS-ratio. A larger number of products and stores leads to more decisions on base stock levels to be made. However, as the model does not include the information as given in the data-driven approach for each product and store there are more possibilities for deviations from GAM. In fact, the model has on average the second lowest total base stock level, with 17.90 % less than GAM.

For the sake of completeness, one should mention that ORR and the sampling method have no significant (linear) influence on the solution quality of the models.

3.6.2.2. Solution Quality

After having determined the most relevant factors for the solution quality, we will now look at the values of ζ_{ij} itself. We group solutions by the number of sample weeks (≤ 10 or > 10), TBO (≤ 6 or > 6) and capacity (≤ 1 or > 1). Within the group, we average the ζ_{ij} values and highlight the best performing model.

Table 3.5.: Average ζ_{ij} of models in instance group

Number of sample weeks	TBO	Capacity	No. inst.	DTW	DPS	GAM	SEQ	SAVS	EDM
≤ 10	≤ 6	≤ 1	29	49.54 %	38.66 %	11.39 %	24.99 %	23.65 %	**10.47 %**
		> 1	36	**7.30 %**	27.74 %	11.25 %	25.95 %	28.63 %	8.99 %
	> 6	≤ 1	28	34.50 %	46.37 %	**8.83 %**	29.60 %	35.79 %	10.06 %
		> 1	22	5.27 %	25.24 %	**3.74 %**	41.85 %	68.50 %	13.64 %
> 10	≤ 6	≤ 1	25	96.53 %	60.28 %	**5.20 %**	14.44 %	19.67 %	14.30 %
		> 1	22	11.40 %	21.88 %	**6.07 %**	32.10 %	29.67 %	10.46 %
	> 6	≤ 1	15	62.28 %	75.40 %	**2.45 %**	23.92 %	25.42 %	4.94 %
		> 1	23	4.53 %	20.90 %	**3.07 %**	32.66 %	60.11 %	7.34 %
		Mean	24	32.42 %	37.87 %	**7.18 %**	27.93 %	35.67 %	10.20 %

In bold: Minimum average ζ_{ij} over all models in the respective instance group

Looking at the highlighted values in Table 3.5, it is interesting to see that the GAM performs best in six of the eight instance groups. Note that the GAM is the best model in all of the four high TBO instance groups, which underlines the findings from before. Also, in all of the instance groups with a high number of sample weeks, GAM is the best model. However, in the instance groups where the TBO and number of sample weeks is small, either the DTW model (for a large capacity value) or the EDM model (for a small capacity value) is the best model. In a next step, we want to determine the value of the data-driven approach and the value of coordination. Comparing the differences between EDM and GAM, one can see that GAM outperforms EDM by 3.02 %-points on average and at maximum per instance group average by 9.90 %-points. This difference is the value of including the data-driven approach into the decision making. The SEQ model differs by 20.75 %-points on average from the GAM and is not among the best approaches in any instance group. This further underlines that a combined decision making between the delivery to the stores and the shelf stacking is favorable and that there is indeed a value of coordinated decision making. The DPS differs by 30.69 %-points from the GAM on average, is performing worst of all models and is not among the best models in any instance group, which illustrates how important the consideration of the joint replenishment effect during pattern creation is. The SAVS model performs second worst on average, which leads to the conclusion that a simple heuristic is not capable of capturing the complexity of the problem setting satisfactorily.

3.6.2.3. Delivery Patterns and Cost Structures

The cost structures within the different models show the drivers behind the solution quality (see Figure 3.3; note that capacity violation costs take up around 1 % on average). Taking the GAM model as a reference point, it becomes evident that the share of the major and minor setup costs, as well as the out-of-stock costs, is much lower than in the other approaches. This underlines that the good performance of the GAM in terms of costs is primarily driven by a reduction of the major and minor setups while reducing the out-of-stock situations.

3.6.3. Case Study

The data of our case study is based on actual point of sales (POS) data of a retailer gathered over 532 consecutive working days (plus 14 bank holidays where stores are closed) in the time between March 2011 and December 2012 at 66 stores. The stores are operating on 6 weekdays (Monday through Saturday) from 8 am to 8 pm. We cluster the hourly POS data and introduce three intra-periods per day, thereby accounting for the morning (hours 8 am through 11 am), afternoon (hours 12 am through 3 pm) and evening (hours 4 pm till last) sales. The demand fluctuates between days in a week and from one week to the next. With, for instance, 28 % of the total weekly demand on one day and 9 % on another, the seasonality is not negligible (see Figure 3.4).

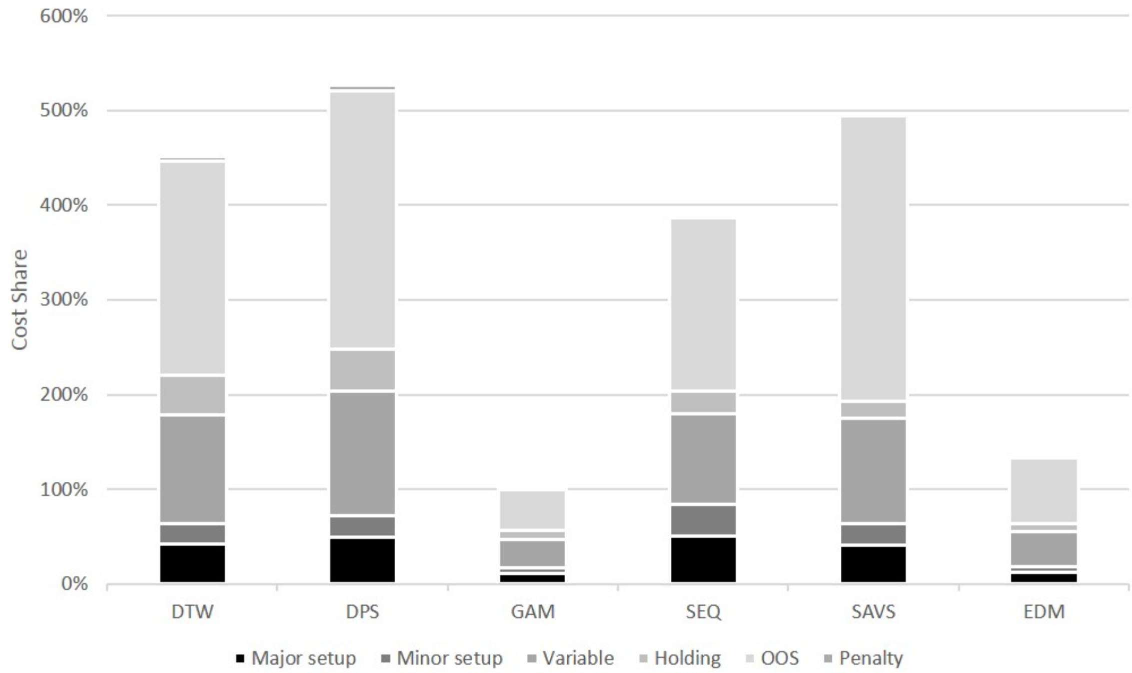


Figure 3.3.: Average percentage cost split (with GAM model as reference)

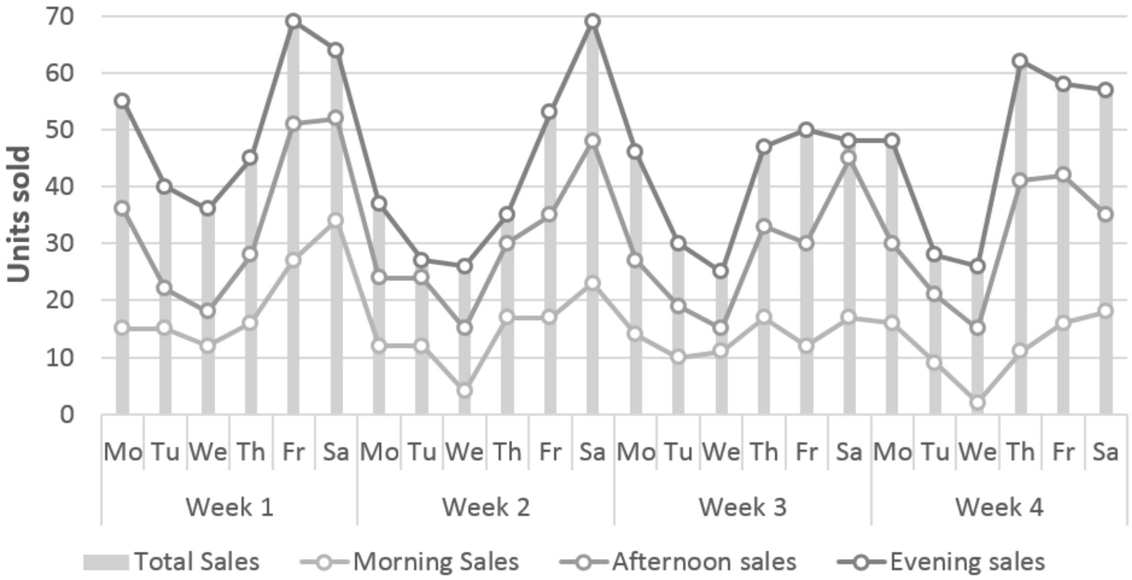


Figure 3.4.: Case study: Demand data of one product for morning (08.00 am - 11.59 am), afternoon (12.00 am - 03.59 pm) and evening hours (04.00 pm - Closing) over four weeks

We apply the GAM, SAVS and EDM approaches to optimize the large dataset. INT, DTW, and DPS are not suitable for this large dataset, as out-of-memory issues occur. As a benchmark for comparison, we include the EDM model to estimate possible cost benefits and exclude SEQ because of its limited performance in the controlled experiment. We set the time horizon to $\bar{T} = 18$. Cost parameters are derived as in Section 3.6.1 based on a TBO equal to 6 (i.e. an order every other day) and MSR = ORR = 0.5. In order to reflect the high complexity of the case study, time restrictions are set to $3,600$ seconds per stage (and $7,200$ seconds for the EDM) or until at least one solution has been found. We include all 66 stores and focus on the 10 products for which most data are available. We split the available data into an optimization dataset of four and a simulation dataset with 26 weeks of data. Based on an assumed weighted average cost of capital of 10 % p.a. broken down to our periods of roughly 1/3 of a day, we set the holding costs per period to $8.7 \cdot 10^{-5}$ times the sales price. We vary capacities as well as the OOS-ratio to obtain the results shown in Table 3.6.

Table 3.6.: Results of ζ_{ij} for the case study

Capacity (β)	OOS-ratio	GAM	SAVS	EDM
150 %	10	0.19 %	26.20 %	**0.00 %**
	20	0.25 %	39.51 %	**0.00 %**
100 %	10	**0.00 %**	26.37 %	2.91 %
	20	**0.00 %**	39.30 %	5.30 %
50 %	10	**0.00 %**	40.40 %	46.39 %
	20	**0.00 %**	51.74 %	69.43 %
	Mean	**0.07 %**	37.25 %	20.67 %

In bold: Best ζ_{ij} value over all models

As before, the GAM performs best on average. Although the EDM gains a small advantage in instances where capacities are large, the GAM is better by 20.60 %-points than the EDM approach on average over all instances and never more than 0.25 % away from the best solution in all instances. SAVS performs worst on average, which is the same result as in the earlier numerical study. However, it obtains better solutions than EDM in the last two instances with tight constraints on capacity.

3.7. Concluding Remarks

We proposed a data-driven method for setting delivery patterns and order-up-to levels for retail stores with stochastic non-stationary demand, considering the handling effort incurred in various processes. The solution methodology can be used for retailers to automatically construct and assign delivery patterns for their stores on a tactical time horizon. Our approach explicitly considers the handling effort in the decision-making process and optimizes this cost-intensive part of the supply chain. We applied a MILP formulation on a three-echelon joint replenishment problem. Due to the computational complexity of the problem, we used decomposition techniques to achieve good results in a fairly short time by splitting the decision making into the construction of delivery patterns in the first and their selection per product and store in the second stage sub-problem.

It has been shown that a genetic algorithm is favorable in the numerical study on fictitious data saving 3.02 % against a deterministic approach with safety stocks. The genetic algorithm is suitable for the large data sets we typically encounter in a real case, where, under varying parameter settings, logistics costs improvements for the 10 products case of 20.60 %-points on average against a deterministic approach with safety stocks is realized. This amounts to an average absolute cost saving of 121.70 Euro per week over 10 products and, if linearly scaled to a full year and full assortment of 2,000 products, it adds up to over 1.3 million Euro.

A limitation of the data-driven approach lies in a possible overfitting. The data does not allow for extrapolation of realizations that might, e.g., lie in the tails of the true underlying distribution. To counteract this effect to at least some extent, the usage of sampling approaches other than Monte-Carlo (e.g., descriptive sampling of Saliby, 1990) can be used.

For further research, looking at the handling effort at the store and warehouse, one might also want to include e.g. the effect of the amount that can be directly filled into the shelf versus the amount that has to be stored in the backroom afterwards (double handling of products). The relationship between the product location in the warehouse and the store, as well as its implications on the handling effort, might also be an interesting field. The next step regarding the data-driven approach would be to include more available data and thus move the approach to a "Big Data" application, as has been done in Ban and Rudin (2014). Further explanatory input variables can be information on promotions, weather, price, etc.. These factors can be used to explain the uncertain demand, while simultaneously making the operational decisions on delivery pattern and order-up-to level as above.

Chapter 4.

Data-driven Retail Inventory Management with Backroom Effects

The backroom effect (BRE) constitutes the handling effort of a replenishment that does not fit on the shelf of a retailer. This effect needs to be included in the decision making of inventory policy parameters as it influences the handling effort, which constitutes a major part of the retailer's operational cost. We propose a mixed integer linear program to calculate the parameters of a periodic review (s, c, S, nq) policy while considering the BRE. The (s, c, S, nq) policy triggers an order when inventory drops below the reorder point s. Also, an order is triggered whenever the inventory drops below the can-order point c, provided at least one other product's inventory level is below s and thus ordered. The order then comprises the smallest integer number n of case packs with size q that brings the inventory level to or above S. As retailers face stochastic non-stationary demand, a data-driven approach based on historical data is applied to this Joint Replenishment Problem. The numerical study shows that including the BRE into the optimization can lead to cost savings with a median of 0.96 % compared to neglecting its effects. Considering the stochasticity in the decision making, cost improvements with a median of 53.23 % have been realized against an approach that only considers average daily demands and a safety stock. The advantage of an (s, c, S, nq) order policy over an (s, S, nq) policy is shown by median savings of 17.99 %.

4.1. Introduction

Retail stores have inventory at two locations: on the shelves and in the backroom. When a replenishment order arrives at the store, it is used to directly re-stack the shelves. Obviously, sometimes the replenishment order does not fit on the shelf, as the shelf space capacity is limited, especially if case pack size and shelf space are not well aligned.

Hence a part of the replenishment quantity (the overflow inventory) has to be stored in the backroom. However, having backroom inventory has two main disadvantages. First, costs arise as the inventory has to be monitored, managed and double handled in order to restock the shelves when free shelf space is available (Eroglu et al., 2013; Sternbeck & Kuhn, 2014). Second, operational complexity increases due to misplaced items in the backroom or in the store and due to inventory record inaccuracies, i.e. discrepancies between inventory recorded in the system and physical inventory (see Corsten & Gruen, 2003; DeHoratius & Ton, 2009; Raman et al., 2001; Ton & Raman, 2010; Waller et al., 2008). All of these reasons make the retailer's decision-making process more complex. Eroglu et al. (2013) call this phenomenon the *backroom effect (BRE)*.

They found out that the misalignment of case pack size, shelf space capacity and inventory control policy is the reason for the BRE. According to the results of the numerical simulation carried out by the aforementioned authors, ignoring the BRE and hence the dynamics between sales area and backroom leads to inefficient replenishment policies and consequently to increasing costs. However, as the case pack size is defined by the manufacturer and the determination of shelf space capacity is a relatively long-term decision, the retailer can only adjust the short-term inventory control policy in order to reduce the total cost. Although the backroom effect has a crucial impact on the cost and efficiency of in-store inventory systems in retail, only a small part of the literature takes its effects into account (Pires et al., 2015).

The *joint replenishment problem (JRP)* determines whether various products are ordered jointly or not, although ordering might not be optimal for each individual product (Aksoy & Erenguc, 1988; Khouja & Goyal, 2008). To minimize the total cost of the inventory system, the ordering decision in a multi-product inventory problem mainly depends on two cost components: the major ordering (or setup, fixed) cost and the minor ordering cost. Major ordering costs arise when an order is placed independently of the number of different items in the order. Minor ordering costs are incurred for every individual item in the replenishment order.

Whenever a retail store is served by a distribution center, major setup costs can be characterized by the transport costs of sending a truck to the store, independent of its utilization. The minor setup costs, which occur per order line along with the variable cost per unit, resemble the shelf stacking effort in retail stores (Curşeu et al., 2008; Van Woensel et al., 2013). The transport costs have been tackled by, e.g., solving inventory routing problems to combine more stores in a truck delivery so as to achieve a better utilization (see, e.g., Gaur & Fisher, 2004; Holzapfel et al., 2016). These approaches model a single-item problem by assuming that each product is served in each delivery with an amount equal to the (deterministic) demands until the next delivery. They neglect the potential of ordering some of the products in different orders more or less often, which could reduce the minor setup costs. Curşeu et al. (2008) and Van Woensel et al. (2013) also use a single-item view of the problem, allowing for batch ordering in order to reduce the minor setup costs. However, major setup costs, like the transport costs depicted above, are neglected. Therefore, the formulation as a joint replenishment problem allows a minimization of both types of fixed costs (major and minor) simultaneously. Balintfy (1964) evaluated and compared classes of multi-item inventory problems and determined a simple decision rule. He introduced the "can-order" point and the (s, c, S) policy. Whenever an order for a particular item has to be issued, i.e., the stock of any item has dropped to the reorder point s, the inventory level of the rest of the items will be checked, and all items which are in the range between can-order point c and reorder point s are ordered jointly (Silver et al., 1998).

Retailers may order multiples of certain batch sizes (case packs) rather than single units. However, the respective effects have long been ignored in academic literature (Waller et al., 2008). We include the consideration of case packs and adjust the traditional (s, c, S) policy, which we will further call (s, c, S, nq) policy. In this policy, whenever an order is triggered by s or c, the smallest integer multiple n of the case pack size q that brings the inventory position above S is ordered. This is somewhat similar to the (s, S, nq) model proposed by Van Woensel et al. (2013), although in their scenario n takes the largest value so that the inventory position after ordering is still smaller than S. We, however, follow the dominant literature on (R, s, nq) policies (e.g., Broekmeulen et al., 2017; Silver et al., 1998), where orders are made to result in an inventory position above s. It is straightforward to see that both interpretations can be easily implemented in our model approach and that the resulting values for S can be carried over from one to the other. For a given value S of our interpretation, using $S + q - 1$ as the order-up-to level in the policy of Van Woensel et al. (2013) will lead to the same policy.

Another feature in retail is the use of delivery patterns (Sternbeck & Kuhn, 2014). These patterns determine at what periods a delivery to the store might be made on an item-level. Although it is beneficial to optimize these patterns with regard to the resulting handling effort (Holzapfel et al., 2016; Taube & Minner, 2017a), this is rarely the case in practice, where delivery patterns are determined by rule-of-thumb in discussions with store managers. We, therefore, suggest the application of an (s, c, S, nq) policy. Based on the categorization of retail planning problems in Hübner et al. (2013), we focus on the interface between distribution and sales, while assuming that strategic decisions on the store design (e.g., infrastructure and layout) and on the distribution setup (e.g., the physical distribution structures), as well as the operational decisions of transport planning (e.g., route planning), are out of scope. Our setting is thus situated at the mid/short term master planning between distribution (regarding the inventory policy) and in-store activities (regarding the in-store handling efforts).

Although most publications dealing with the determination of suitable policy parameters assume demand to be stationary over time, this is clearly not the case in retail (see Brijs et al., 1999; Broekmeulen & Van Donselaar, 2004; Ehrenthal et al., 2014; Gaur & Fisher, 2004; Martel et al., 1995; Van Donselaar et al., 2010, 2006). Even so, for existing approaches to work, it is necessary to either know future demand in advance or make distributional assumptions or forecasts based on historical sales data. Iyer and Schrage (1992) developed a concept for using historical demand data to directly generate inventory control parameters, i.e. an order policy, for use in future inventory control. The advantages of this *data-driven* (or direct, robust, distribution-free) approach are that no distributional assumptions and parameter estimations are needed and that it can be applied to a non-stationary stochastic demand setting. The drawback of the approach is that historical demand data can only form samples and an infinite number of samples would be needed to fully cover the true unknown demand distribution. However, using samples in a clever way by applying descriptive sampling (Saliby, 1990) can mitigate this drawback and produce good results.

Van Zelst et al. (2009) revealed that 38 % of the operational logistical costs in retail are attributed to the in-store handling of products (or items). Hence, it is necessary to include these costs when determining inventory policy parameters in order to increase the operational efficiency (Reiner et al., 2013). Van Donselaar et al. (2010) show that retail managers override order suggestions of software solutions that do not account for the resulting handling effort anyway.

The handling effort in retail is rarely considered in the decision making and in mathematical models for inventory problems (Van Woensel et al., 2013). Furthermore, there is hardly any literature dealing with the backroom effect and with the subsequent implications for optimal inventory policies. This chapter contributes to the literature on the backroom effect in retail, data-driven approaches, and the joint replenishment problem while considering substantial characteristics of the retail industry: non-stationary stochastic demand and ordering in multiples of a given case pack size. The following questions, in particular, are discussed in this chapter: Is it advisable to include the BRE in the optimization process? Is the application of the data-driven approach promising? When is it beneficial to consider an (s, c, S, nq) policy over an (s, S, nq) policy? In order to answer these questions, a mixed integer linear program (MILP) with a data-driven approach for the joint replenishment problem is developed. The backroom effect is included in the MILP. Additionally, a cyclical approach is introduced to determine the initial inventory as part of the optimization process. The mathematical models are validated in a numerical study assuming stochastic non-stationary demand.

This chapter is organized as follows: a literature review is given in Section 4.2. Section 4.3 captures the model formulation and various valuation models that focus on the data-driven approach, the backroom effect or the joint replenishment problem. A numerical study is conducted in Section 4.4. A brief summary of the findings and an outlook on further research is given in Section 4.5.

4.2. Literature Review

4.2.1. (s, c, S) Policies

To calculate the optimal inventory control parameters with respect to an (s, c, S) policy, Balintfy (1964) developed an algorithm based on the multiple machine queuing theory. Other approaches based on Markov Decision Processes to determine the parameters for periodic review settings have been developed by Ohno et al. (1994) in exact and Johansen and Melchiors (2003) in approximative ways. Extending the general problem, further publications deal explicitly with demand correlation between items. H. Feng et al. (2015) consider the structure of the optimal policy of a corresponding Markov Decision Process and formulate heuristics capturing the demand correlation. Tsai et al. (2009) use an association clustering algorithm to decompose a large set of items into groups that experience correlation. They then adapt a heuristic from Silver (1974) to obtain the parameters of the (s, c, S) policy. Although we do not explicitly consider correlation, our data-driven approach does not imply any assumptions on the underlying demand distribution, which is why possible demand correlations are implicitly considered in the determination of the policy parameters. While the above-mentioned publications allow general (stationary) demand distributions, most other publications resort to Poisson or compound Poisson demand distributions (Goyal & Satir, 1989; Khouja & Goyal, 2008).

In general, publications dealing with (s, c, S) policies assume backorders rather than lost sales for any unfulfilled demand. Lost sales joint replenishment problems are scarce (Bijvank & Vis, 2011) and lost sales (s, c, S) policies under stochastic demand have not been investigated. Nagasawa et al. (2015) deal with lost sales and (s, c, S) policies, but only under dynamic deterministic demand. Their heuristic approach based on a genetic algorithm focuses on finding good values for the can-order level while assuming that values for s and S are set exogenously. Nonetheless, their MILP formulation of the problem has some similarities with our model in Section 4.3, although it neglects uncertain demand and the simultaneous optimization of all policy parameters. None of the above-mentioned literature on (s, c, S) policies includes batch sizes (i.e., case packs) for order quantities. However, case packs, the lost sales environment and non-stationary stochastic demand are dominating characteristics in retail (Hübner et al., 2013), which is why the existing approaches cannot capture our problem setting.

Non-stationary stochastic demand has been dealt within a joint replenishment context by applying the three heuristic strategies of static uncertainty, dynamic uncertainty and static-dynamic uncertainty of Bookbinder and Tan (1988). The first strategy generates decisions for the whole planning horizon and implements all of those prior to its start. The second strategy generates and implements decisions only for each period individually along the planning horizon. In a mixture of both, the third strategy generates decisions for the whole time horizon before the start, but only implements the first period and updates the remaining decisions on a rolling horizon basis. Martel et al. (1995) and Hua et al. (2009) both use the static-dynamic uncertainty approach for a constrained multi-item non-stationary demand setting. They suggest a stochastic programming approach, which results in multiple static demand sub-problems. Both papers assume that the plan does not need to be robust, but decisions can be revised at any time, which is not applicable in our setting, where we want to keep the parameters of the inventory policy robust. Also, minor setup costs are neglected. Tempelmeier and Hilger (2015) apply the static uncertainty approach, focusing on robust solutions. A MILP model to solve the stochastic multiple-item capacitated lot-sizing problem is formulated. The model includes β service level constraints and setup carry-overs. They also propose a fix & optimize heuristic, where smaller sub-problems are solved and the results are merged to obtain the solution of the original problem. Taube and Minner (2017a) (see also Chapter 3), use a data-driven approach for a retail-based non-stationary demand environment in a three-echelon system. They assume an (R, S) policy where R constitutes the delivery pattern and S the order-up-to level. Both policy parameters are determined by a data-driven MILP model. As opposed to this, this chapter incorporates case pack sizes and the BRE into the decision making and uses the (s, c, S, nq) policy.

4.2.2. Backroom Effect in Retail

Eroglu et al. (2013) determined an optimal (r, Q) replenishment policy while considering the BRE. In their approach, the BRE is triggered by the overflow inventory. In order to calculate the BRE cost, they determine the expected amount of overflow inventory per order cycle (for a corrected derivation of the expected overflow term, see Atan & Erkip, 2015) and multiply it with a fixed unit BRE cost. They assume that the overflow costs are linear with the number of items. Their numerical study shows that ignoring the BRE has negative effects on operational costs. Furthermore, a misalignment of case pack size, shelf space and inventory control policy are causes of the BRE. In contrast to Eroglu et al. (2013), we focus on an (s, c, S, nq) replenishment policy instead of an (r, Q) policy while considering the backroom effect. We do so in order to capture the joint replenishment problem. Another important contribution of our work is the inclusion of the calculation of the inventory control parameters in the optimization process, i.e. in the mixed integer linear program. Hence, the determination of the replenishment policy is not based on the economic order quantity model, as in Eroglu et al. (2013).

Van Woensel et al. (2013) model (s, S, nq), (s, nq) and a mixture of both policies in a retail setting where only batches (i.e. case packs) can be ordered. They consider handling efforts and model the problem as a Markov decision process. They, however, focus on a stationary single-item problem. In a similar setting, while considering equidistant delivery patterns and case packs, Van Donselaar and Broekmeulen (2013) determine safety stocks in a lost sales environment and an (R, S, nq) policy under a target fill rate. They improve an approximation of the fill rate by regression analysis to reduce the approximation error.

In other single-item approaches, Sternbeck (2015) and Kuhn et al. (2015) demonstrate the huge impact of the misalignment of case pack sizes when optimizing those, taking the handling effort and backroom effect into account. Both assume an (R, s, nq) policy, while Kuhn et al. (2015) optimize the relevant policy parameters along with the case pack sizes by modeling a Markov decision process. Recently, Broekmeulen et al. (2017) determine the location of unpacking a case pack (either in the warehouse or in the store) and their underlying inventory policies. Those policies are (R, s, nq) for unpacking in store and (R, s, S) for unpacking in the warehouse. The effects of both alternatives on cost and especially handling effort are captured and single-item decisions are made. They assume that customer demand is stationary in time, which does not reflect the actual situation in retail (Ehrenthal et al., 2014; Taube & Minner, 2017a).

4.2.3. Data-driven Approach

Iyer and Schrage (1992) applied the so-called *data-driven* approach to find optimal control parameters for the deterministic (s, S) policy with a polynomial time algorithm. The idea of this approach is to avoid the distribution fitting step and instead use historical data samples as direct input for the optimization of the inventory control policy. Note that the classical approach is to choose a theoretical distribution that describes the historical sales data properly, then to estimate suitable parameters and thus fit the theoretical distribution to the sales data. Hence, the major advantage of the direct approach is that there is no risk of an erroneous determination of the inventory control parameters due to wrong distributional assumptions. Using the data-driven approach, historical demand streams provide multiple samples of the target time horizon (e.g. several weeks). Therefore, the optimized inventory control parameters are valid for all samples. For example, assuming a reorder level of 20 on Mondays and a demand stream representing 10 samples (i.e. demand data of 10 weeks), the inventory level on all 10 Mondays is examined and if the considered inventory level is less than 20, an order is triggered on the respective Monday.

Sachs and Minner (2014) investigate the newsvendor problem under unobservable lost sales and Beutel and Minner (2012) determine safety stocks, both publications using a data-driven approach. Taube and Minner (2017a) use the same approach to determine delivery patterns and order-up-to levels with a mixed integer linear program based on stochastic non-stationary demand. Further inventory planning problems are solved by Bertsimas and Thiele (2006) with a robust linear programming formulation. They also derived closed-form expressions for key parameters that define the optimal policy to provide a deeper insight into the way uncertainty affects the optimal policy. Bertsimas and Thiele (2014) provide a tutorial on how uncertainty sets can be obtained for robust optimization using historical data. Other applications of data-driven approaches include, e.g., Huh et al. (2011) for inventory control of censored demand data and Bertsimas and Doan (2010) for call center staffing. To tackle overfitting in models using big data as an input, Ban and Rudin (2014) introduce several machine learning methods and apply a big data-driven newsvendor model to a nurse staffing problem.

4.3. Model Formulation

4.3.1. Assumptions

One retail store that is served by one warehouse with ample stock is considered. There is one in-store backroom serving the shelves and storing overflow inventory. Arriving order quantities in excess of available shelf capacity cause variable (minor) and fixed (major) overflow costs. Ordering incurs major, minor and variable costs as in the general JRP. Available stock in the store at the end of the period incurs holding cost and lost sales cost occurs for any customer demand of a product the inventory level of which is not sufficient. Products are assumed to be non-perishable.

As input for the model, a set V of samples representing different weeks of data are used, whereby demand is assumed to be non-stationary over time and stochastic. The data samples comprise historical demand data of commonly purchased products in a retail store. An influence on future demand due to the retailer (e.g. advertising, rebates, promotions, etc.) or due to lost sales is excluded from the model. Also, trends in the demand series are not explicitly considered. Order lead time, i.e. the elapsed time between the placement of an order and its delivery, is zero, which means that the replenishment order arrives before demand occurs. We assume that an order is placed after the final inventory has been updated. Note that this final inventory is equal to the beginning inventory of the next period. The order arrives the next period before the first demand occurs (e.g. via overnight shipment).

4.3.2. Sequence of Events

One sample $v \in V$ consists of several periods $t \in T$ and in the beginning of each period t, the inventory level I_{ivt} of an item $i \in I$ is updated. Note that I_{ivt} is also defined for $t = |T| + 1$, i.e., for one additional period, to reflect the final inventory at the end of the time horizon. Throughout the chapter, a sample v refers to one week and hence a period t signifies a specific day within this sample week. If an order is triggered due to the inventory control policy, i.e. the inventory level I_{ivt} of a specific item i is below the reorder point s_{it} or can-order level c_{it}, the order quantity is determined such that the inventory is at least replenished to a predefined order-up-to level S_{it}.

The order incurs major setup costs SC, minor setup costs of sc_i per product and variable ordering costs of p_i per item. As the order lead time is zero, the delivery immediately arrives at the store and is initially placed on the shelf and, if the shelf space capacity cap_i is reached, in the backroom. This process causes a major overflow cost K per overflow event. Furthermore, each item that is replenished to the shelf incurs a minor overflow cost k_i per unit. The two cost factors reflect the generally non-linear handling costs in retail (Broekmeulen & Van Donselaar, 2004; Curşeu et al., 2008; Sternbeck & Kuhn, 2014; Van Zelst et al., 2009). We assume that replenishments from the backroom are made instantaneously so that any customer demand d_{ivt} is satisfied while inventory in the store is still available. Unmet demand l_{ivt} is denoted as a lost sale and incurs a penalty cost ls_i per unit. As capital is tied up with store inventory (shelves + backroom), holding costs h_i per unit on hand arise at the end of each period t.

4.3.3. Calibration and Evaluation Based on a Data-driven Concept

The general concept for finding optimal inventory control parameters and subsequently evaluating the replenishment policy based on the data-driven approach is shown in Figure 4.1.

For the calibration, i.e. the calculation of the inventory control parameters, samples representing multiple weeks are used as deterministic and direct input for the optimization. Again, note that, as mentioned before, no distributional assumptions are made

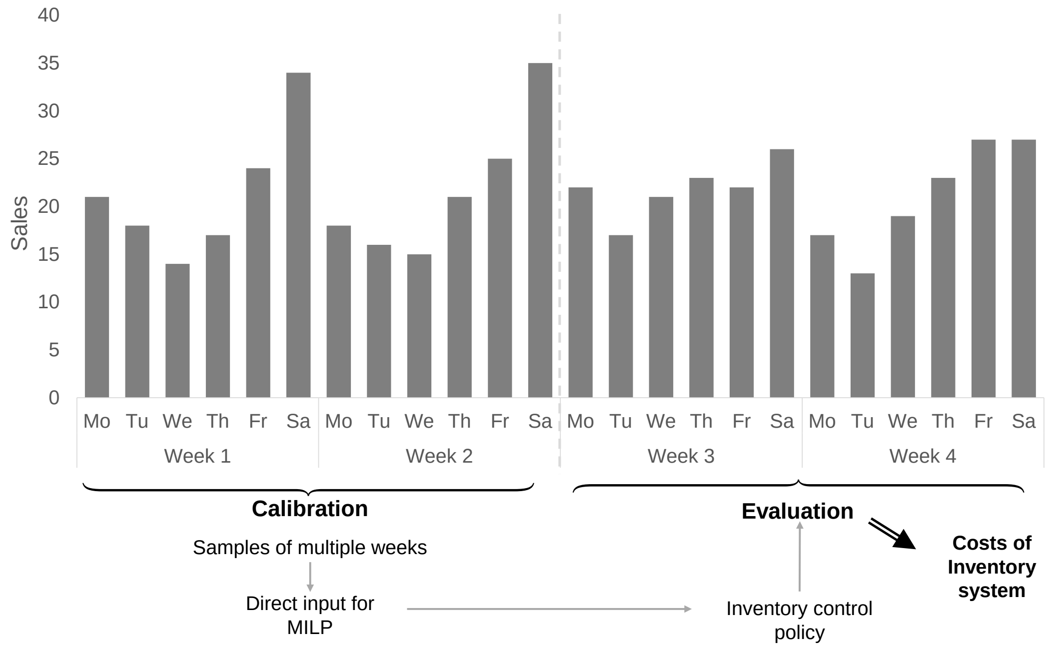


Figure 4.1.: Example of calibration and evaluation based on a data-driven concept

and no demand distribution parameters are estimated. After the optimization process, the determined parameters of the (s, c, S, nq) order policy are used for future inventory management. During this process, which is called the evaluation, the inventory control parameters are fixed and no revision is possible. At the end of the evaluation process, the inventory system's costs are obtained. This is done to ensure that the resulting average weekly costs are estimated within a confidence bound. In practice, the assumption of no revision can obviously be relaxed if the approach is implemented in a rolling horizon fashion.

4.3.4. Mixed Integer Linear Program (MILP)

The decision variable I_{ivt} captures the available store inventory (i.e. shelves + backroom) for item i at the beginning of period t of sample v. The binary decision variable y_{ivt} indicates whether or not an order is triggered by the reorder point s_{it}. Orders take the quantity $n_{ivt} \cdot q_i$ if the binary order variable $u_{ivt} = 1$. Here, n_{ivt} is an integer denoting the ordered number of case packs with size q_i. A binary decision variable Y_{vt} is introduced in order to denote a major ordering event, while the aforementioned variable u_{ivt} indicates a minor ordering event, regardless of whether this order has been triggered by the must-order (s) or can-order point (c). Overflow inventory upon delivery arrival is denoted by the decision variable w_{ivt}. The binary decision variable Z_{vt} indicates whether a major overflow cost due to overflow inventory upon arrival in period t of sample v is incurred ($Z_{vt} = 1$) or not ($Z_{vt} = 0$). The satisfied demand of item i in period t of sample v is captured by the decision variable b_{ivt}, and lost sales are depicted by the decision variable l_{ivt}. The actual number of periods is contained in set T. The determined reorder point s_{it}, can-order point c_{it} and order-up-to level S_{it} of item i in period t are valid for all samples. This ensures non-anticipativity of the approach, as the decisions are valid for all uncertain outcomes and not individually revised in each sample.

Note that, throughout this chapter, the following MILP formulation is referred to as the *full* model. In some of the constraints in the model, a sufficiently large number M_i is needed. In order to be efficient and to enhance the performance of the model, this number is set to the maximum of total demand over all samples, $M_i := \max_{v\in V} \left\lceil \sum_{t\in T} d_{ivt}/q_i \right\rceil \cdot qi$.

For an overview of the introduced notation see Table 4.1.

$$\begin{aligned} \min \sum_{i\in I}\sum_{v\in V}\sum_{t\in T} (p_i \cdot n_{ivt} \cdot q_i + sc \cdot u_{ivt} + k_i \cdot w_{ivt} + ls_i \cdot l_{ivt}) \\ + \sum_{v\in V}\sum_{t\in T} (SC \cdot Y_{vt} + K \cdot Z_{vt}) + \sum_{i\in I}\sum_{v\in V}\sum_{t\in T} h_i \cdot I_{iv(t+1)} \\ + \sum_{v\in V}\sum_{i\in I} |T| \cdot h_i \cdot \sum_{t\in T} (n_{ivt} \cdot q_i - b_{ivt}) \end{aligned} \tag{4.1}$$

subject to

$$0 \leq \sum_{t\in T} (n_{ivt} \cdot q_i - b_{ivt}) \leq M_i \qquad \forall i \in I, v \in V \tag{4.2}$$

$$d_{ivt} = b_{ivt} + l_{ivt} \qquad \forall i \in I, v \in V, t \in T \tag{4.3}$$

$$I_{iv(t+1)} = I_{ivt} + n_{ivt} \cdot q_i - b_{ivt} \qquad \forall i \in I, v \in V, t \in T \tag{4.4}$$

$$I_{ivt} - s_{it} + 1 \leq M_i \cdot (1 - y_{ivt}) \qquad \forall i \in I, v \in V, t \in T \tag{4.5}$$

$$s_{it} - I_{ivt} \leq M_i \cdot y_{ivt} \qquad \forall i \in I, v \in V, t \in T \tag{4.6}$$

Table 4.1.: Notation for the full model

Parameters	
$i \in I$	Items
$t \in T$	Periods
$v \in V$	Samples
SC	Major setup costs
sc_i	Minor setup costs of item i
k_i	Variable overflow costs of item i
K	Major overflow costs
p_i	Variable costs of item i
h_i	Holding costs for item i
ls_i	Out-of-stock costs of item i
cap_i	Shelf space capacity of item i
q_i	Casepack size of item i
d_{ivt}	Customer demand for item i in sample v and period t
M_i	Sufficiently large number for item i

Decision variables	
s_{it}	Reorder level of item i and period t
c_{it}	Can-order level of item i and period t
S_{it}	Order-up-to level of item i and period t
I_{ivt}	Inventory level of item i at the beginning of period t in sample v
b_{ivt}	Satisfied demand for item i in sample v and period t
l_{ivt}	Unmet demand for item i in sample v and period t
n_{ivt}	Integer order quantity of casepacks for item i in sample v and period t
y_{ivt}	Binary trigger of reorder level s_{it} for item i in sample v and period t
u_{ivt}	Binary trigger of minor ordering event for item i in sample v and period t
Y_{vt}	Binary trigger of major ordering event in sample v and period t
w_{ivt}	Overflow quantity for item i in sample v and period t
Z_{vt}	Binary trigger of major overflow event in sample v and period t

$$n_{ivt} \leq \frac{M_i}{q_i} \cdot u_{ivt} \qquad \forall i \in I, v \in V, t \in T \tag{4.7}$$

$$u_{ivt} \leq Y_{vt} \qquad \forall i \in I, v \in V, t \in T \tag{4.8}$$

$$Y_{vt} \geq y_{ivt} \qquad \forall i \in I, v \in V, t \in T \tag{4.9}$$

$$Y_{vt} \leq \sum_{i \in I} y_{ivt} \qquad \forall v \in V, t \in T \tag{4.10}$$

$$I_{ivt} - c_{it} + 1 \leq M_i \cdot (1 - u_{ivt}) \qquad \forall i \in I, v \in V, t \in T \tag{4.11}$$

$$c_{it} - I_{ivt} \leq M_i \cdot (1 - Y_{vt} + u_{ivt}) \qquad \forall i \in I, v \in V, t \in T \tag{4.12}$$

$$I_{ivt} + n_{ivt} \cdot q_i \geq S_{it} - M_i \cdot (1 - u_{ivt}) \quad \forall i \in I, v \in V, t \in T \tag{4.13}$$

$$I_{ivt} + n_{ivt} \cdot q_i \leq (S_{it} + q_i - 1) + M_i \cdot (1 - u_{ivt}) \quad \forall i \in I, v \in V, t \in T \tag{4.14}$$

$$w_{ivt} \geq \big(I_{ivt} + n_{ivt} \cdot q_i - cap_i\big) - M_i \cdot (1 - u_{ivt}) \quad \forall i \in I, v \in V, t \in T \tag{4.15}$$

$$w_{ivt} \leq M_i \cdot Z_{vt} \qquad \forall i \in I, v \in V, t \in T \tag{4.16}$$

$$S_{it} \geq s_{it} + 1 \qquad \forall i \in I, t \in T \tag{4.17}$$

$$S_{it} \geq c_{it} + 1 \qquad \forall i \in I, t \in T \tag{4.18}$$

$$s_{it} \leq c_{it} \qquad \forall i \in I, t \in T \tag{4.19}$$

$$n_{ivt} \in \mathbb{N}_0 \qquad \forall i \in I, v \in V, t \in T \tag{4.20}$$

$$y_{ivt}, Y_{vt}, u_{ivt}, Z_{vt} \in \{0, 1\} \qquad \forall i \in I, v \in V, t \in T \tag{4.21}$$

$$s_{it}, c_{it}, S_{it}, w_{ivt}, b_{ivt}, l_{ivt} \geq 0 \qquad \forall i \in I, v \in V, t \in T \tag{4.22}$$

$$I_{ivt} \geq 0 \qquad \forall i \in I, v \in V, t \in \{T, |T| + 1\} \tag{4.23}$$

The objective function (4.1) includes the variable, minor and major ordering, minor and major overflow, lost sales and holding cost. The holding costs are calculated based on $I_{iv(t+1)}$, which represents the final inventory of period t. The third row captures the penalization of items ordered in excess of the sample's total demand. We do so because the cyclic formulation implies that this excess inventory is never used during the time horizon and thus incurs holding costs for the whole time horizon. To determine the initial inventory of item i, i.e. the inventory level in the beginning of the first period of the sample v, a cyclical approach is introduced by (4.2). By this approach, the items ordered in excess of the total demand of the sample have to be larger than or equal to zero. Thus, the initial inventory can only be smaller than or equal to the ending inventory of the sample. (4.3) ensure that satisfied demand plus lost sales equal the total customer demand. The calculation of the inventory at the end of period t, which is the inventory in the beginning of period $t + 1$, as mentioned before, is given in (4.4). By this inventory balance constraint, the inventory in the beginning of period $t + 1$ is defined as the sum of initial inventory level and order quantity less the satisfied demand in period t. (4.5) and (4.6) determine whether or not an order is placed because the inventory level is below the reorder level s. (4.7) places the upper bound on the order quantity to either 0 if $u_{ivt} = 0$ or M_i/q_i otherwise, thus representing the maximum of total demanded case packs over all samples. In case of an order placement, (4.8) and (4.9) indicate a major ordering event. Note that items can only be jointly replenished if and only if an order of at least one item is already scheduled in the respective period due to (4.10). (4.11) and (4.12) indicate whether this specific product is going to be replenished if the inventory level is below the can-order level c_{it} and an order is placed anyway ($Y_{vt} = 1$). Whenever a product is replenished, the ordered amount, which is an integer multiple of the case pack size, has to result in an inventory position equal to or greater than the order-up-to level, but less than the order-up-to level plus one case pack size; otherwise the order quantity is zero as stated in (4.13) and (4.14). (4.13) forces the order quantity to raise I_{ivt} over S_{it}, while (4.14) ensures that I_{ivt} does not exceed $S_{it} + q_i - 1$. Upon delivery, potential overflow inventory that does not fit on the shelf is determined by (4.15). If overflow inventory arises in a period, a major overflow cost is denoted by (4.16). To guarantee that the order quantity is always greater than zero in the case of a replenishment, (4.17) and (4.18) ensure that the order-up-to level is strictly greater than the reorder and can-order level. (4.19) assures that the can-order level is at or above the reorder level s. At last, integer variables including the number zero are defined in (4.20), binary variables are depicted in (4.21) and nonzero variables are listed in (4.22) and (4.23).

Remark: The formulation can be extended to consider a constant positive lead time $L < |T|$. To do so, the net inventory I_{ivt} has to be calculated by exchanging n_{ivt} in (4.4) with $n_{it\lambda(v-L)}$, where $\lambda(a) = \begin{cases} a & \text{if } a \leq 0 \\ |T| - a & \text{if } a < 0 \end{cases}$. Due to our cyclic approach, the period shift operator λ is needed, as it allows orders from the previous cycle to arrive in the current cycle. The inventory position can then be defined as $I_{ivt} + \sum_{r=1}^{L-1} n_{it\lambda(v-r)} \cdot q_i$, which needs to be exchanged with I_{ivt} in (4.1), (4.5)–(4.6) and (4.11)–(4.14).

4.3.5. Valuation Models

In order to quantify the performance of the full model and the respective inventory control policy, additional valuation models are formulated. A model that does not take the costs of overflow inventory into account during the optimization process (*NoBRE*), a model that is based on a single sample of expected daily demands and a safety stock (*NoDD*) and a model that does not consider the joint replenishment problem and uses an (s, S, nq) policy (*NoJRP*).

4.3.5.1. Model Without Backroom Effect (NoBRE)

As the BRE may have a significant influence on the performance of an inventory control policy, a valuation model has been developed in order to investigate to what extent the implementation of this effect enhances the performance of the full model and the corresponding (s, c, S, nq) policy.

The difference between this and the full model of Subsection 4.3.4 is that $NoBRE$ does not consider the BRE, i.e. the overflow inventory and the respective costs, during the optimization process. k_i and K are set to zero, the decision variables w_{ivt} and Z_{vt} and constraints (4.15) and (4.16) concerning the overflow inventory are excluded from the model. However, the backroom costs are determined in the post-processing in order to calculate the total cost caused by the replenishment policy during evaluation.

4.3.5.2. Model Without Data-driven Approach (NoDD)

By this valuation model, the inventory control policy is determined without the data-driven approach. In contrast to the data-driven approach, which uses the data of several weeks as direct input for the optimization, the demand input for $NoDD$ is now expressed for one week based on averages over all samples. Therefore, before the optimization process, the mean demand $\bar{d}_{it}$ of item i in period t is calculated as

$$\bar{d}_{it} = \frac{1}{|V|} \cdot \sum_{v \in V} d_{ivt} \qquad \forall i \in I, t \in T.$$

$\bar{d}_{it}$ is used instead of d_{ivt} in the model formulation, while all other decision variables and constraints are treated as being applied in a setting with one sample. Additionally, we introduce

$$I_{i1t} \geq \gamma \cdot \bar{d}_{it} \qquad \forall i \in I, t \in T. \tag{4.24}$$

(4.24) is a safety stock constraint, which ensures that at least a given multiple γ of each period's demand is available in the inventory at the beginning of each period.

Furthermore, this approach is compared to the following safety stock constraint

$$I_{i1t} \geq \gamma \cdot q_i \qquad \forall i \in I, t \in T. \tag{4.25}$$

(4.25) ensures that at least a given multiple γ of the case pack size of a product is available in inventory at the beginning of each period.

Note that, in the following, the model including (4.24) is referred to as $NoDD(1)$ and the model including (4.25) as $NoDD(2)$.

4.3.5.3. Model Without Joint Replenishment (NoJRP)

To quantify the possible cost savings due to a joint order policy, a valuation model that only determines a reorder level s_{it} and an order-up-to level S_{it} as inventory control parameters has been developed, the result of which is an (s, S, nq) policy. Thus, the order policy provided by this model only schedules a joint replenishment by chance if it is optimal to order for each item individually.

The objective function in this case does not consider orders triggered by a can-order level c_{it}. Consequently, (4.11) - (4.12) and variable c_{it} of the full model have been removed.

4.4. Numerical Study

The different models are investigated and compared with each other in order to answer the following questions: Is it advisable to include the BRE in the optimization process? How much can performance increase by considering stochasticity via the data-driven approach? In what cases does the (s, c, S, nq) outperform the (s, S, nq) policy? Moreover, the problem characteristics are varied to observe potential influences on the performances of the different approaches.

All models for the calibration and all algorithms for the evaluation have been implemented in FICO Xpress IVE (Version 7.9). A 64-bit system with an Intel Xeon CPU with 2.60 GHz and 8 cores was used to conduct the experiments.

4.4.1. Data Instances

To create test data, we first fix the time horizon to six periods, representing a week of six working days from Monday to Saturday. Then we set the mean demand μ_i for each item i as a uniform random value of the interval $[1, 100]$ and for the case pack size q_i of the interval $[1, 32]$. We determine the standard deviation by the *coefficient of variation (CV)*, which we vary over the instances, with $CV = 0.3, 0.5, 0.7$. Based on this, the standard deviations σ_i for all items can be calculated as $\sigma_i = \mu_i \cdot CV$.

Additionally, in order to represent seasonality within the week, seasonal factors f_t for each period t are considered in the data generation process. These factors are deduced by using averages of information about sales seasonality from different publications (see Table 4.2). A general formula to calculate the seasonal factor of a specific item, given the daily mean demand over all periods μ and the period related mean demand μ_t, is given by $f_t = \frac{\mu_t}{\mu}$.

Table 4.2.: Comparison of daily sales fraction in retail literature (in %)

Day of the week	Mon	Tue	Wed	Thu	Fri	Sat
Van Donselaar et al. (2010)	11.0	11.0	13.0	18.0	27.0	20.0
Van Donselaar et al. (2006)	12.0	11.0	12.0	16.0	26.0	23.0
Gaur and Fisher (2004)	12.7	12.5	13.5	16.7	18.3	26.4
Brijs et al. (1999)	14.0	13.0	13.0	19.0	22.0	19.0
East et al. (1994)	10.3	11.3	12.4	19.6	24.7	21.7
Kahn and Schmittlein (1989)	14.1	14.3	15.3	17.8	19.7	18.8
Rounded average	*12.3*	*12.2*	*13.2*	*17.9*	*22.9*	*21.5*

Data estimated from graphics or tables in respective publications.

To reflect the varying values for $\mu_{it} = \mu_i \cdot f_{it}$ and σ_i and the necessity to consider discrete distribution functions to generate the demand, we apply the approach of Adan et al. (1995). This approach chooses a suitable discrete distribution function (F_{it}) that matches the first two moments for every item i and period t.

Having these distribution functions, we draw random demand values based on two different sampling approaches: Monte-Carlo sampling and Descriptive sampling. In the Monte-Carlo sampling, we draw a uniform random value from the interval $[0, 1]$ for each sample v, i.e. $\mathcal{Z}_{ivt}^M \sim U[0, 1]$. In the Descriptive sampling approach of Saliby (1990), we proceed for each item i and period t individually. We first create $|V|$ values $\mathcal{Z}_v^D$, representing equally spaced points on $[0, 1]$, i.e. $\mathcal{Z}_v^D = \frac{v}{|V|+1}$. We use the deterministic set $\mathcal{Z}^D$ and randomize the sequence, i.e., we shuffle those values, for each item i and t separately, which results in $\mathcal{Z}_{ivt}^D = \mathcal{Z}_{\Theta(v)}^D \quad \forall i \in I, t \in T$, where $\Theta(v)$ is the shuffling operator. Having both, $\mathcal{Z}_{ivt}^M$ and $\mathcal{Z}_{ivt}^D$, we create the individual demand values by using the inverse transformation method, i.e. we get the smallest value of the distribution function, larger than or equal to $\mathcal{Z}_{ivt}$, i.e., $d_{ivt} = \arg\min\{x = 0, 1, 2, \ldots : F_{it}(x) \geq \mathcal{Z}_{ivt}\}$.

Remark: To operationalize the sampling approaches in practice, one would use the point of sales data for all products and periods as datasets $\mathcal{D}_{it}$ of the daily demand. For both approaches, $\mathcal{Z}_{ivt}^M$ and $\mathcal{Z}_{ivt}^D$ are constructed as explained above. Then the datasets $\mathcal{D}_{it}$ have to be ordered by the demand values in ascending order. From these ordered sets, the values corresponding to $\mathcal{Z}_{ivt}$ are found in a similar way as in the inverse transformation method by finding the xth entry that covers a cumulated percentage of the historical demands that is at least as high as $\mathcal{Z}_{ivt}$. Thus, $d_{ivt} = \arg\min\{x = 0, 1, 2, \ldots : x/|\mathcal{D}_{it}| \geq \mathcal{Z}_{ivt}\}$, where $|\mathcal{D}_{it}|$ is the cardinality of set $\mathcal{D}_{it}$.

Eroglu et al. (2011) show that retailers in practice often express the shelf space for a product as a multiple of the case pack size and that they use a typical shelf-space-to-case-pack ratio of 1.5. Hence, we set the shelf space size cap_i to $1.5 \cdot q_i$ in our numerical study.

To determine the values of major and minor setup costs, we use the average fix costs per order A as a parameter, which we vary by setting $A = 5, 10$ in the experiment. Based on A, the minor and major setup costs can be derived. In order to get there, we have to consider the ratio between major and minor set-up cost (MSR), defined as another parameter of the numerical study. We set

$$sc = (1 - MSR) \cdot A$$

and

$$SC = MSR \cdot |I| \cdot A,$$

where $|I|$ represents the cardinality of set I, i.e. the number of products.

The major overflow costs are set to $K = SC$. For all other remaining cost values we sample from a normal distribution with mean values adapted from Broekmeulen et al. (2017) and a CV of 0.2. Thus, we set the mean holding cost to $h = 0.25$ Euro/year $= 0.0007$ Euro/day, the mean lost sales penalty to $ls = 0.275$ Euro and the mean variable order cost and mean variable overflow cost to $p = k = 0.03$ Euro.

Lastly, we vary the safety stock parameter γ for both model variants without the data-driven approach ($NoDD(1)$ and $NoDD(2)$) between the values $1, 2, \ldots, 10$.

We summarize the parameter variation in Table 4.3, which takes place in a full factorial design over all models introduced before (i.e., $Full$, $NoBRE$, $NoDD$, $NoJRP$).

Table 4.3.: Parameter variation

Parameter	Values		
Number of items $\lvert I\rvert$	5	10	15
Number of samples $\lvert V\rvert$	10	20	30
CV	0.3	0.5	0.7
A	5	10	
MSR	0.3	0.6	
Sampling method	Monte-Carlo	Descriptive	
Safety stock factor γ (only NoDD)	$1, 2, \ldots, 6$		

In addition to this full factorial design, runs with 50 and 100 items are done in order to give insight into the scalability of our approach. For those runs, we set the average fix costs per order A to 10, the number of samples to 30, the coefficient of variation CV to 0.5 and the ratio between minor and major setup cost MSR to 0.3. As sampling method, we use Descriptive sampling.

4.4.2. Results

The solver has a time limit of $3,600$ seconds for the full-factorial numerical study and has a time limit of 24 hours for the large instances with 50 and 100 items. First, the full and valuation models are executed in order to determine inventory control parameters for each instance during calibration. In order to compare the performance of the inventory control policies, the respective inventory system's average weekly costs are simulated during the evaluation phase. The evaluation is done on a large number of additional Monte-Carlo generated samples, to receive results, which are unbiased by a potential over-fitting during calibration. To guarantee the significance of the received solutions, the number of simulated weeks (sample size ρ) for each instance is determined such that a 95 % ($1 - \alpha = 0.95$) confidence interval based on the standard normal distribution $\mathcal{N}(0,1)$ (as all sample sizes ρ are > 31) is formed with a tolerance of $\epsilon = 1$ % (Law, 2015). For the large instances, we specified a maximum number of simulated weeks of 20,960 for tests with 50 items and of 10,480 for tests with 100 items.

4.4.2.1. Overall Performance

To give the reader a first impression, we provide the 25 % quantile, median, mean and 75 % quantile of the relative cost increase regarding the valuation models against the full model over all instances in Table 4.4. For the *NoDD* model, we show the values for the best γ value as is later investigated in Section 4.4.2.3.

Table 4.4.: Cost increase of all valuation models against the full model (in %)

	NoBRE	*NoDD*(1) *(best)*	*NoDD*(2) *(best)*	*NoJRP*
25 % quantile	−3.36	31.66	40.01	9.45
Median	0.96	53.23	54.92	17.99
Mean	1.54	55.95	58.43	20.85
75 % quantile	6.25	76.52	73.65	29.64

One can assess that including the BRE, using the data-driven approach and using the JRP approach have in common that they all have a positive influence on the median. Note that, as we can see in the 25 % quantile for *NoBRE*, there are instances where the full model does not improve the outcome. This indicates that the performance of the full model is dependent on the instance characteristics. Note that this refers to the out-of-sample performance. In the subsequent sections, we will investigate those outcomes in more detail and explain the influences on the performance of the full model.

4.4.2.2. Impact of the Backroom Effect

In order to answer the question if it is promising to include the BRE in the optimization process, Figure 4.2 shows the median inventory system's cost differences caused by neglecting the backroom effect (model *NoBRE*).

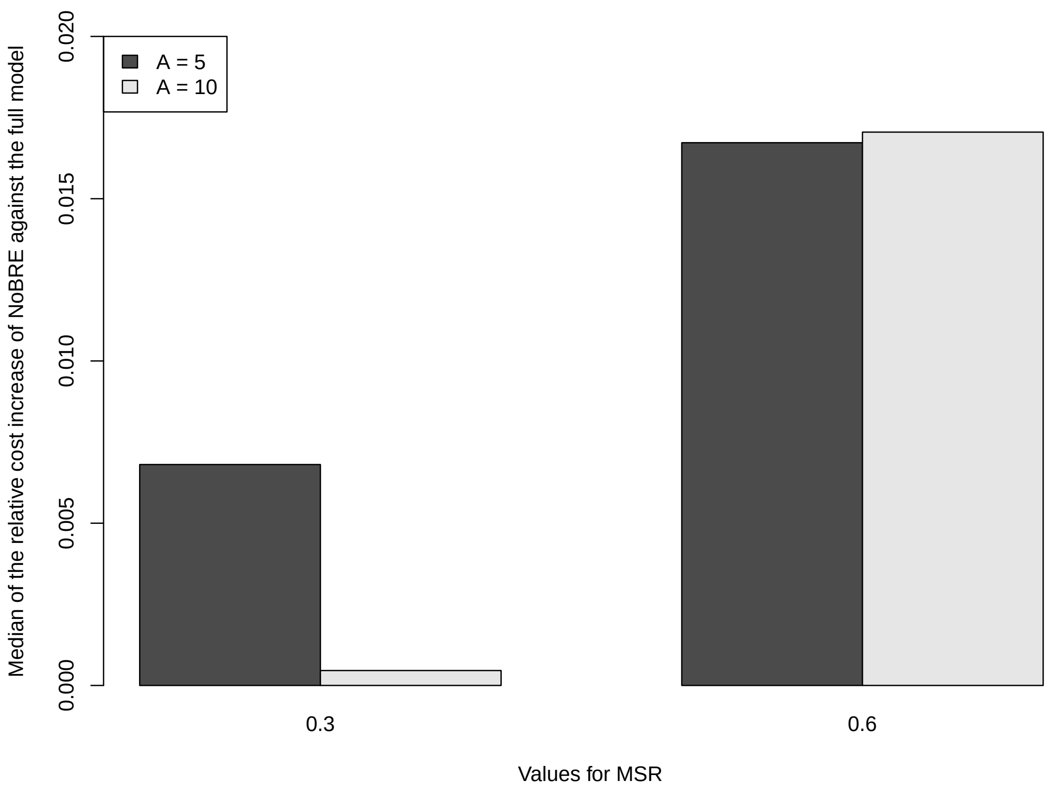


Figure 4.2.: Median of the cost increase due to the BRE

The first observation is the small cost increase of the $NoBRE$ model for $MSR = 0.3$ and $A = 10$. In this setup, the minor setup cost is as high as possible and, thus, the advantage of joint product orders fades as products are more likely ordered when it is optimal for them. As a result, the products are frequently stored directly on the shelves and less often in the backroom.

A general increase of the cost benefit for the full model can be observed when MSR rises. A rise in this value leads to higher major setup and in turn higher major overflow costs, as $K = SC$. As the full model is able to reduce the overflow occurrences in comparison with $NoBRE$, also the savings of the full model against $NoBRE$ regarding the overflow costs and subsequently the relative cost difference rises with a rising MSR value.

4.4.2.3. Impact of the Data-Driven Approach

Regarding the data-driven approach, we first show the impact of the safety stock level to be set. In Table 4.5 and Table 4.6, the 25 % quantile, median, mean and 75 % quantile are shown for various levels of the safety stock factor γ.

Table 4.5.: Cost increases of $NoDD(1)$ depending on γ (in percent)

	γ									
Quantile	1	2	3	4	5	6	7	8	9	10
25 %	31.66	34.03	43.44	53.83	47.34	67.36	78.90	87.52	94.33	101.47
Median	53.23	54.48	64.18	75.83	62.97	79.31	88.81	97.24	105.50	112.81
Mean	55.95	57.89	63.82	70.38	61.25	78.80	89.22	98.38	105.26	112.48
75%	76.52	76.75	83.40	94.02	78.32	89.91	99.44	110.30	118.67	126.39

Table 4.6.: Cost increases of $NoDD(2)$ depending on γ (in percent)

	γ									
Quantile	1	2	3	4	5	6	7	8	9	10
25%	40.01	44.57	38.05	36.79	37.41	42.68	45.28	49.02	51.31	55.53
Median	54.92	64.39	57.82	56.09	56.46	65.10	67.95	70.61	72.62	77.16
Mean	58.43	70.69	64.93	64.80	65.07	69.26	72.34	76.30	77.83	82.56
75%	73.65	93.53	89.13	88.78	89.99	91.76	94.52	99.40	100.12	105.60

Observing the impact of γ, the best results are achieved with a value of $\gamma = 1$ in both cases. And comparing both tables, it seems that $NoDD(1)$ renders slightly better results in this case when looking at the median and the mean. But even then, the median of the cost increase lies at 53.23 %, thus we can observe a huge cost impact. Another interesting observation is the variance in the cost impact across all γ values. This variance can be explained by the cost trade-off of the fixed γ for all products. As γ is not product-specific, the safety stocks are likely to be too high for some and too low for some other products. As γ is varied, the resulting lost-sales costs are impacted.

In addition, the holding costs incurred by safety stocks also have an impact on the joint replenishment, as products might be ordered less (more) often by the model when γ changes. This cost trade-off is individual for every product and hence the cost behavior in regard to γ is non-monotonic.

When looking at Table 4.6, we see that the variance of the cost impact across all γ values is far smaller. In this case, the safety stocks are more adapted to the products as the γ values are tied to the case pack size. This, hence, has a remarkable influence on the variance of the cost impact.

We analyze the drivers of this cost difference in Figure 4.3, where we show the median cost increase of the best *NoDD* model ($NoDD(1)$ with $\gamma = 1$) in relation to the number of samples used and the value for the average fix cost A.

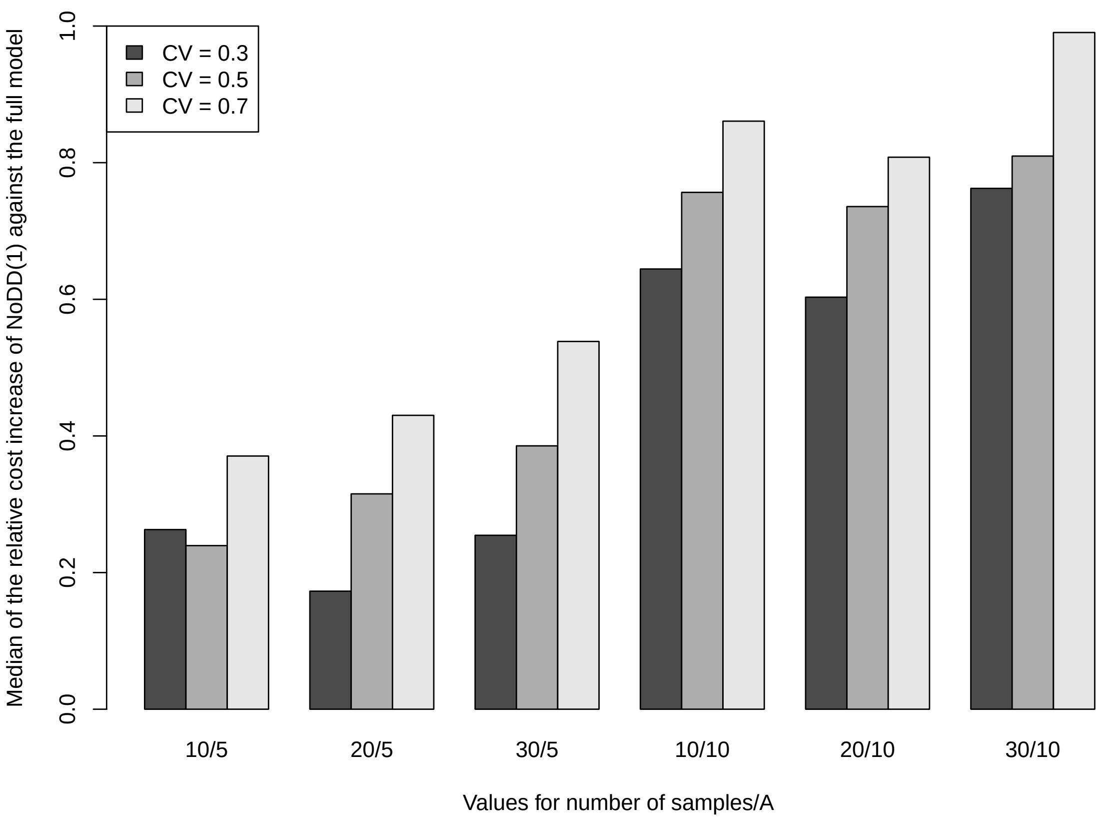


Figure 4.3.: Median of the cost increase of the best *NoDD* setup ($NoDD(1)$ with $\gamma = 1$)

As expected, the cost increase by the $NoDD$ model is, in general, reinforced by the number of samples. With a larger number of samples the full model is able to incorporate more information regarding the true demand distribution into the decision making than the $NoDD$ model, which is not able to use that information as it is only using the mean demand values per period. Hence, major cost savings can be realized with a high number of samples. A similar rise in cost can be observed for an increase in the value for A. An investigation of the order behavior of the $NoDD$ model reveals that order fixed costs and thus the number of orders are much higher than in the full model. As information regarding the samples is missing, the parameters of the (s, c, S, nq) policy are not set in such a way that they would appropriately minimize the fixed order costs. This goes hand in hand with the findings regarding the $NoJRP$ model in the next section. Basically, a larger CV also yields a larger cost benefit for the data-driven model. This is due to the fact that the information on the mean loses value for cases with a large CV. In those cases, demands representing the tails of the distribution occur more often, which have a huge cost impact, especially due to the resulting lost sales.

4.4.2.4. Impact of the Can-order Policy

Figure 4.4 shows the cost differences between the (s, c, S, nq) policy of the full model and the (s, S, nq) policy of the *NoJRP* model, again in relation to the number of samples used and the value for the average fixed costs A.

A can-order-policy offers advantages over an (s, S, nq) policy, which increases with A, the number of products and, in general, with an increasing number of samples. In order to be able to render a benefit from the (s, c, S, nq) approach, joint product orders have to be possible. This is why an increasing number of products also shows an increase in costs for the $NoJRP$ model. Furthermore, the (s, c, S, nq) primarily aims at achieving a saving in the fixed costs due to the JRP characteristics. Hence, a larger value for A promises a bigger cost saving potential. The full model needs more samples in order to provide good parameter values, which is logical as it needs to optimize one more parameter than the (s, S, nq) model. This is why savings are the largest if the number of samples is high.

Another interesting observation can be made when we look at the overflow and order setup costs of $NoJRP$ and compare them to the full model. We observe two effects. As expected, major setup costs rise within $NoJRP$ by a significant amount (94.8 %), as the

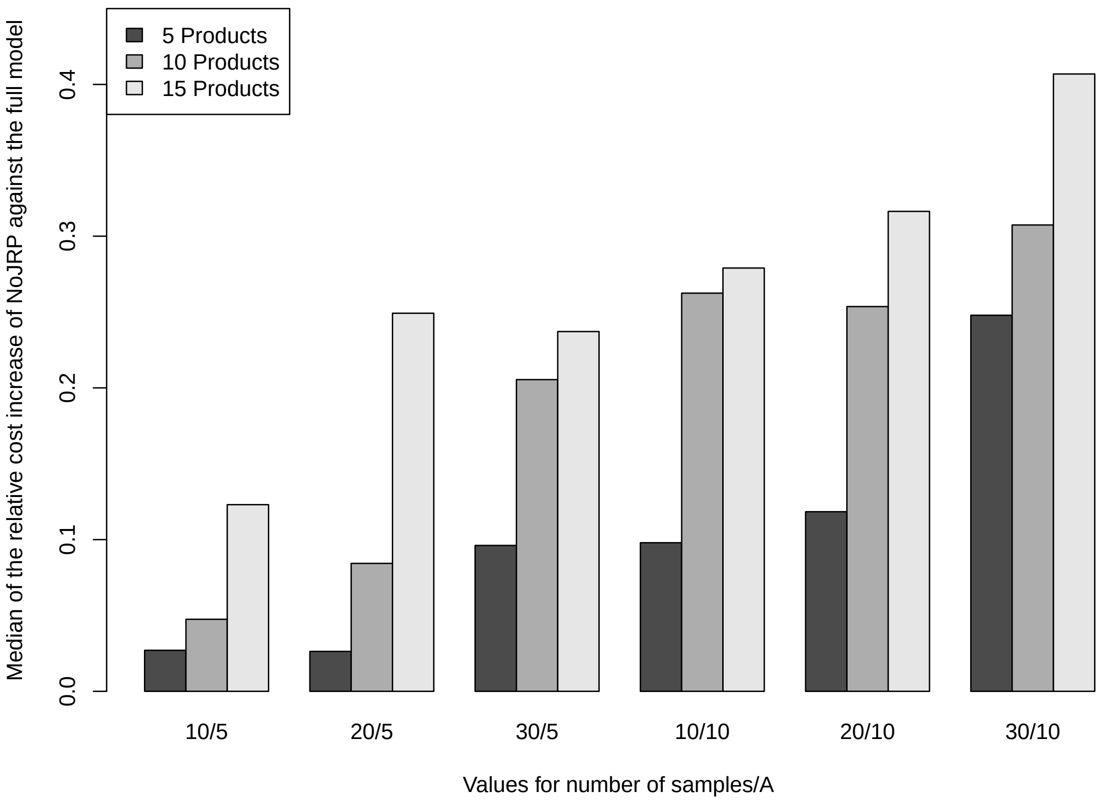


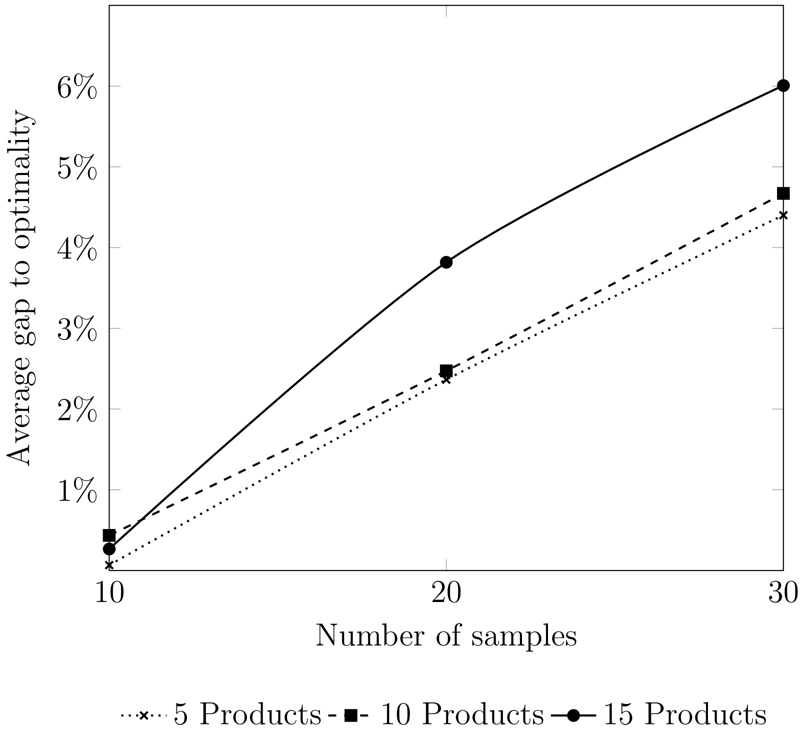
Figure 4.4.: Median of the cost increase of NoJRP

approach does not coordinate the ordering between the products. On the other hand, minor overflow costs are reduced by 0.7 %. The larger minor overflow in the (s, c, S, nq) model can be explained by the fact that orders are advanced for some products in the can-order policy, which might exceed the allocated shelf space on an item-level. However, major overflow costs are again 94.94 % higher in the (s, S, nq) model, because there is, again, no coordinating effect. In total, overflow costs are 37.7 % larger in $NoJRP$. By focusing on the JRP, we reduce both major setup and major overflow costs. $NoBRE$ already reduces the major overflow costs by 46.9 % over the results of $NoJRP$, although it does not consider the BRE during optimization.

4.4.2.5. Optimality Gap

Figure 4.5 shows the average optimality gap of the full model within the time limit of 3,600 seconds for the solver and across all instances of the full-factorial numerical study.

Figure 4.5.: Average optimality gap of the full model



It is quite clear that the solver needs more runtime as the problem complexity increases. Obviously, the average optimality gap rises with a larger number of samples since the full model has to incorporate more information regarding the true demand distribution into the decision making. An increase in the number of products also leads to a higher optimality gap, because the solver has to determine more parameters. However, optimality gaps rise linearly rather than exponentially in the number of samples and products. For instance, for 30 samples, when we triple the number of products from 5 to 15, the optimality gap only rises from an average of 4.4 % to 6 %. This makes the model attractive for the application to large-scale settings.

4.4.2.6. Scalability of the approach

In order to gain insight into the scalability of the proposed approach, additional runs with 50 and 100 items were executed. The results of the runs are depicted in Table 4.7, which shows the optimality gap of the full model and the cost increases of all valuation models against the full model.

Table 4.7.: Optimality gap of full model and cost increases of valuation models against the full model for the large instances (in %)

Products	Optimality Gap *Full Model*	$NoBRE$	$NoDD(1)$ *best* ($\gamma = 2$)	$NoDD(2)$ *best* ($\gamma = 4$)	$NoJRP$
50	8.55	11.93	72.81	90.24	46.76
100	11.72	9.74	73.94	60.82	40.58

As we can see, the full model performs significantly better than before, in almost all cases, when we compare the results of the median of the cost increases shown in Table 4.4. Especially the cost increases of the $NoBRE$ and $NoJRP$ model are remarkable. However, these results are what we expected since, with an increasing number of products, the operational complexity of coordinating orders rises and the (s, c, S, nq) approach can display its advantages. As already described in one of the previous sections, an increase of the assortment of a retailer is beneficial to the (s, c, S, nq) policy because more products can be jointly ordered and, hence, major setup costs can be saved.

Moreover, the optimality gap only moderately increases. When the number of products is doubled from 50 to 100, the optimality gap increases only by 3.2 %-points.

4.5. Conclusion

In this chapter, a data-driven approach is applied in order to calculate inventory control parameters for retail stores. The demand is assumed to be stochastic and non-stationary. Furthermore, with the introduction of the backroom effect into the decision making, the double handling of goods in the store is considered. Finally, a new approach is provided to calculate the parameters for the (s, c, S, nq) inventory control policy. To eliminate assumptions on the size of the initial inventory, a cyclical approach is included in the mathematical model, which sets the initial inventory level as part of the optimization process.

The numerical study showed that considering the backroom effect leads to cost savings with a median of 0.96 %. Furthermore, it is evident that including the stochasticity into the problem solving is a necessity, as cost reductions with a median of 53.23 % are achieved when we apply the data-driven approach as opposed to optimizing the policy based on just one sample representing average daily demands, even if safety stocks are assumed. Further outcomes of the numerical study are the implications derived from using the (s, c, S, nq) can-order policy, rather than an (s, S, nq) policy. Improvements in the inventory management performance with a median of 17.99 % are possible.

Moreover, we observed that the approach can cover a large number of products because the complexity, which is indicated by the average optimality gap at termination, rises linearly.

As an interesting research field, the effect of including the determination of the case pack size and the shelf space capacity in the optimization process of the retailer could be worth investigating. The consideration of the backroom effect in the mathematical model while assuming non-linear variable overflow cost, as assumed in Sternbeck and Kuhn (2014) might also render promising results. Another worthwhile approach might be to introduce the product location into the mixed integer linear program, as it has an influence on the handling effort and hence is part of the backroom effect. Furthermore, food retail struggles with perishability of products, which in turn has a huge effect on the best order policy in the joint replenishment problem (Kouki et al., 2016) and might be dependent on consumer preference behavior (Amorim et al., 2014). Also for this kind of products the inclusion of the backroom effect into the decision making should be taken into account.

One limitation of our approach is the assumption that replenishments from the backroom to the shelf happen instantaneously. As Corsten and Gruen (2003) point out backrooms are poorly organized in reality and thus it is likely that products are not found or replenishments to the shelf are scheduled at later times of the day when a stock-out on the shelf already occurred earlier. One could include these effects in the model formulation by defining intra-day schedules for in-store replenishments (see Taube & Minner, 2017a) or reduce the amount of overflow inventory to be usable for demand by a certain probability of availability on the shelf.

Chapter 5.

Resequencing mixed-model assembly lines with restoration to customer orders

We consider a supplier who delivers modules to an original equipment manufacturer (OEM) just-in-time and just-in-sequence. Production at the supplier is done via a mixed-model assembly line. The time between knowing the OEM sequence and delivering the finished workpieces to the OEM is small. Nonetheless, resequencing for the mixed-model assembly line at the supplier might be advantageous under various objectives such as workload balancing, leveling of materials consumption or color batching. However, if resequencing is done, the effort to restore the original OEM sequence should be small to achieve this in time. We propose a model for optimizing resequencing under the condition that restoring the original sequence is achieved via a first-in-first-out (FIFO) strategy, where workpieces are stored in mix banks at the end of production and only the workpieces at the front of those banks have to be dispatched in order to rebuild the original sequence. The model is the combination of an assignment or traveling salesman and a vehicle routing problem. We adapt the load balancing, material leveling, and color batching problem from the sequencing literature to our formulation and present numerical results derived from a controlled testbed. They show that, compared to producing the OEM sequence as-is, huge savings in the objective values (> 50 % on average), are made. Furthermore, a limited look-ahead approach leads to good solutions in just a small number of seconds, even for large scale problems.

5.1. Introduction

In order to cope with the ever increasing number of product variants, while maintaining high output volumes, mixed-model assembly lines have prevailed to allow for mass-customization. In mixed-model assembly lines, the sequencing of workpieces has a huge impact on the flow through the assembly system (Boysen et al., 2009b). Mixed-model assembly lines in general (see Boysen et al., 2009a) and the corresponding sequencing problem in particular (see Boysen et al., 2009b) have been researched quite extensively. The problem of resequencing, i.e., the altering of a given sequence in order to facilitate production, has gained importance, either with the intent of coping with disruptions of the target sequence during production or to actively change the sequence to facilitate production.

The problem analyzed in this chapter deals with (first-tier) module suppliers delivering just-in-sequence (JIS) to their customer (in most cases the original equipment manufacturer (OEM)) by producing on a mixed-model assembly line. As they have to deliver a sequence given prior to production start, this sequence normally also determines the flow through the mixed-model assembly line of the module supplier (Islamoglu et al., 2014).

However, in most cases, flows in the OEM production do not take the production flow of the suppliers into account (Fredriksson, 2006). It would be beneficial to alter this sequence during assembly at the supplier to balance the load, to level material consumption or to reduce the number of setups (e.g., color changes) in the same way as is done with the sequencing decisions made for the OEM assembly. An alteration of the given sequence, on the other hand, leads to the necessity of restoring the sequence given by the OEM after assembly is finished, as e.g., in the box assembly line in Islamoglu et al. (2014). This might require a considerable searching and rearranging effort, while the time between notification of the OEM sequence and delivery to the OEM is minimal (Doran, 2001, 2002; Fredriksson, 2006; Meyr, 2004; Swaminathan & Nitsch, 2007).

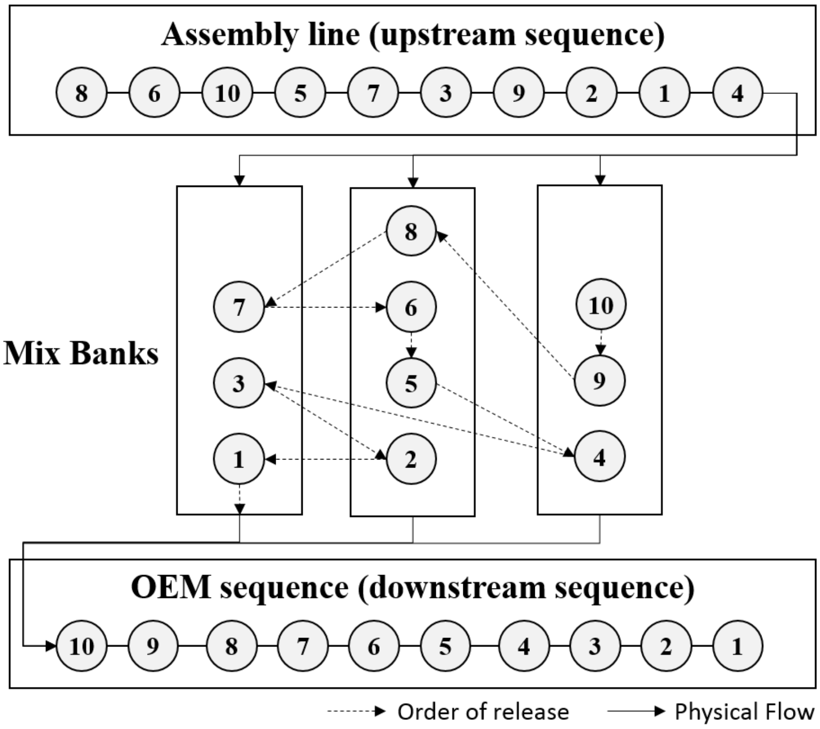
Within this setting we suggest to engage in a resequencing decision to optimize the production flow for the supplier under several objectives. Still, we take into account that sequence restoration has to be efficient to make the approach practically viable.

We motivate our setting and the objectives of load balancing, material leveling and color batching by our experience from several applied projects at different divisions of a first tier automotive supplier producing major modules just-in-sequence for OEMs. Modules, such as, e.g., car seats, are likely to be supplied JIS within a small time frame (e.g., 3 hours production window in Doran (2002), 5 hours information lead time in Swaminathan and Nitsch (2007)), which makes it paramount to keep the restoration of the OEM sequence in a possible resequencing as efficient as possible. Other even more complex applications of JIS supply are given by other module suppliers, with e.g., over $10,000$ cockpit variations for the production of the Smart (Chick et al., 2014; Doran, 2002).

Each module, in general, has a variety of options and a large number of parts, which highly affect assembly operations (Islamoglu et al., 2014) and JIS suppliers are, in general, likely to face varying task times in a mixed model assembly line (see Swaminathan & Nitsch, 2007). That task times and material consumption vary during the module production from one workpiece to the next can also be observed in the examples of the production of a dashboard module in Agnetis et al. (1997) and of front-end modules in Islamoglu et al. (2014). This variance of task times and material consumption during assembly is the basic motivation for the load balancing and material leveling objective. While suppliers might have to deal with different colors in a paint shop, also setup operations in between different product types might take place (e.g., readjustments of machines), both leading to settings like in the color batching objective of our model.

We propose a mixed integer linear program (MILP) model that alters the sequence given by the OEM in order to find a more favorable one for assembly, whilst considering that workpieces are placed on so-called mix banks, from where the original sequence can be restored without much effort by applying a first-in-first-out (FIFO) principle (see Figure 5.1). The system starts with the assembly line, where workpieces are subsequently processed. At the end of the assembly line, a finished workpiece is moved to one of several mix banks. We define mix banks (or selectivity banks, storage lanes, line storage system) as several (parallel) lines, where each of them has a distinguished capacity to store workpieces with workpieces entering and leaving the system on a FIFO basis. While the technical definition does not differ from existing approaches such as (Boysen et al., 2012; Boysen & Zenker, 2013; Choi & Shin, 1997; Spieckermann et al., 2004; Sun et al., 2015), we use the mix banks as a temporal storage in front of the outbound, rather than in front of a subsequent production step (such as, e.g., a paint shop). We implement three of the most common problems in sequencing/resequencing (Boysen et al., 2009b,

Figure 5.1.: Resequencing and sorting example for an OEM sequence 1-2-3-4-...-10



2012), namely the load balancing, the material leveling and the color batching problem.

The first part of the model comprises the sequencing on the assembly line (see the upstream sequence in Figure 5.1) and is modeled either by an assignment problem (AP) or a traveling salesman problem (TSP). We use both formulations as the AP formulation is necessary for the material leveling and the TSP formulation is necessary for the color batching problem. For the load balancing problem, both formulations could be applied, however, the AP formulation yields better performance. We furthermore propose a limited look-ahead heuristic for all model variants to reduce computational complexity and be able to solve large-scale problems.

In order to investigate the behavior of the different model formulations and their savings potential in an operational setting, we set up a numerical study on a controlled testbed. The contribution of this chapter lies in the introduction of a new relevant problem setting, a model formulation adaptable to multiple objectives and a solution approach for solving problems of practice-relevant size. In particular, to support configuration and organizational decisions, we provide insights to the following questions:

- (RQ1): What is the benefit of applying resequencing for the supplier in terms of the three objectives?
- (RQ2): How does the design of the mix banks (number and capacity) influence the achievable savings?
- (RQ3): How well does the limited lookahead approach scale for a setting of practical size?
- (RQ4): How do changes in the operational parameters affect savings, i.e., under what circumstances is a resequencing approach more/less beneficial?

The chapter is structured as follows. In Section 5.2, we give an overview of the relevant resequencing literature. The core module of the model (the sequencing and sorting) and the different objectives are presented in Section 5.3 and tested on a controlled data set in Section 5.4. The chapter concludes in Section 5.5 with a summary and outlook for further research.

5.2. Literature Review

The literature regarding resequencing in mixed model assembly lines has been reviewed by Boysen et al. (2012). Based on their classification, we find our problem setting to be a physical pro-active resequencing with mix banks. The physical resequencing, as opposed to the virtual resequencing, actually shifts workpieces to create a new sequence during assembly. The virtual resequencing, on the other hand, reassigns customer orders to workpieces that are physically identical but does not alter the physical sequence during assembly (e.g., Inman & Schmeling, 2003). As we use the model to determine the best sequence for assembly, we do so pro-actively to achieve a certain goal as we are not only restoring a given sequence that has been distorted before (e.g., Boywitz et al., 2016). Lastly, we are using mix banks (or selectivity banks, storage lanes) as a buffer, which allows the storage and retrieval of workpieces in a FIFO manner on parallel lanes (e.g, Boysen & Zenker, 2013; Boywitz et al., 2016; Choi & Shin, 1997; Ding & Sun, 2004; Spieckermann et al., 2004; Valero-Herrero et al., 2014). Alternatively, pull-off tables extract workpieces from a given sequence and re-enter them into the sequence at a later point (e.g, Boysen et al., 2010; Gujjula & Günther, 2009; Lahmar & Benjaafar, 2007;

Lahmar et al., 2003) and automatic storage/retrieval systems allow for a fully flexible assignment and retrieval of workpieces (e.g, Ding & Sun, 2004; Inman & Schmeling, 2003).

We decide to use mix banks for our approach because of their widespread usage in the industry and their low implementation cost (Spieckermann et al., 2004). Pull-off tables require a more complex logic for moving workpieces into the pull-off table and reinserting them into the sequence at a later point. With mix banks, we only need to decide on which mix bank to put the next workpiece, as the retrieval sequence is given by the smallest index of all workpieces in the front of the mix banks. Also, Boysen et al. (2012) point out that mix banks, in particular, have suffered from considerable neglect in the operational resequencing literature.

The pro-active resequencing carried out to achieve a better flow during assembly has only attained little focus. Most dominantly, publications have dealt with the color batching problem, where sequences that minimize the number of color changeovers in a downstream paint shop are created. Exact approaches based on mixed-integer linear programs were introduced by Spieckermann et al. (2004) and Ding and Sun (2004). Epping and Hochstättler (2003) present a dynamic programming approach inspired from molecular biology. Heuristic approaches regarding rules for placement and retrieval of workpieces can be found in e.g., Sun et al. (2015). The aforementioned publications have in common that mix banks are used, same as in our setting. Other publications regarding the paint shop problem predominantly deal with pull-off tables (e.g, Lahmar & Benjaafar, 2007; Lahmar et al., 2003; Lim & Xu, 2009).

To achieve a more balanced load on the workstations, Franz et al. (2014) present a MILP and heuristic approaches to minimize the number of overload situations. Their problem consists of a set of unblocked orders that have to be inserted into a given segment of an existing sequence, therefore they do not consider the resequencing potential of the existing sequence. A combined optimization of overload situations and the effort for rescheduling workpieces is tackled by Gujjula and Günther (2009). The authors also face JIS constraints, however, from the point of view of the OEM who receives materials from the supplier JIS and thus cannot alter an initially provided sequence too much during the course of production. Also, the rescheduling is done via pull-off tables, rather than mix banks.

The material leveling objective smooths material consumption during production.

Deviations of the cumulated material consumption from a targeted average consumption are to be minimized. Under consideration of limited resequencing capabilities, Boysen et al. (2010) investigate this problem, considering pull-off tables, and present exact and heuristic solution approaches. Based on heuristic rules, Choi and Shin (1997) deal with the material leveling goal (amongst others), when mix banks are in place.

The car sequencing problem predefines rules and minimizes the violations of these rules. These sequencing rules (denoted by $H_o : N_o$) define that a certain attribute o (e.g., a labor intensive operation) is not to be repeated more than H_o times in N_o subsequent cars of the sequence. Boysen et al. (2011) use pull-off tables and Boysen and Zenker (2013) use mix banks as a buffer and provide MILP formulations and a decomposition algorithm to solve the problem. Valero-Herrero et al. (2014) consider the car resequencing problem using mix banks and formulate several rules for placing and releasing workpieces. They evaluate the rules by discrete event simulation.

All of the aforementioned approaches have in common that a given sequence upstream of the buffers is resequenced to produce one new sequence downstream. The problem setting differs to ours as we produce a new sequence upstream of the mix banks and restore the original sequence downstream of the mix banks. Also, in the models above, the consideration of restoring the original sequence does not take place and is in general rarely considered (Ding & Sun, 2004). Yet, considering the restoration of the sequence during resequencing is vital if only limited buffers are available, as otherwise a huge manual sorting effort might happen.

Boywitz et al. (2016) propose a model where a given upstream sequence is restored to a target downstream sequence by using mix banks. As the upstream sequence is given, resequencing capabilities are limited. Thus, workpieces at the end of a mix bank might be blocking the release of the next workpiece of the target sequence. These blocking items then have to be stored in an overflow area. The target of their optimization model is to minimize the necessary overflow area. Note that this chapter differs not only in the objectives treated but especially in the upstream sequence, which we deliberately change while ensuring that the restoration by the mix banks is possible without blocking.

Most of the MILP models with mix banks suffer from symmetry, as each mix bank is represented by dedicated decision variables (i.e., decision variables receive an index for each mix bank). Thus, assigning e.g., the same subsequence alternatively to two different mix banks might generate two separate solutions, however with the same objective. In our approach, the two-index vehicle flow formulation (Toth & Vigo, 2014) for the vehicle routing problem (VRP) allows that mix banks are not explicitly numbered, which reduces the symmetry.

In addition to the above, there are models that show how to optimize several downstream sequences given one upstream sequence (e.g., one paint shop for each mix bank; see Han & Zhou, 2010; Lutfe et al., 2015) or optimize one downstream with several upstream sequences (e.g., Ko et al., 2016). However, neither of the two is applicable to our setting and they are thus not considered further.

Also, suppliers of standardized components with batch production are not included in this chapter. Note that our setting considers JIS supply of finished workpieces after assembly to the OEM, rather than targeting on JIS supply of materials needed for assembly as in e.g., Gujjula and Günther (2009). For a general discussion of the benefits and drawbacks when supplying modules to OEMs, see Fredriksson (2006) and Islamoglu et al. (2014) and for JIS production of suppliers, see Doran (2001).

To summarize, our problem setting is unique in that we resequence upstream of the mix banks to allow for a better flow during assembly while ensuring a restoration of the original sequence downstream with the entire process being constrained by the flexibility provided through the use of mix banks.

5.3. Mathematical Models

5.3.1. Assumptions and Parameters

We consider the production of a sequence of length n and use indices i and j to indicate the position of a workpiece in the original sequence ($i, j = 1, \dots, n$) of the OEM. This sequence is assumed to be known and deterministic before the production starts with the first workpiece of the sequence. There is one assembly line and, although the sequence might be altered during assembly, the original sequence has to be restored at the end. To do so, we assume a number of m mix banks, each having a capacity of f workpieces. A finished workpiece at the end of the assembly line is subsequently moved to the last position of one of the mix banks. Restoration of the original sequence has to be done by pulling workpieces FIFO from within each mix bank. The release of a workpiece from the mix bank is allowed only when it is the next in line, i.e., the workpiece with the smallest index of all workpieces not yet released. It must potentially be possible to temporarily place the whole sequence on the mix banks, i.e., $f \cdot m \geq n$ to ensure a feasible solution of the VRP formulation. As Boysen and Zenker (2013) point out, this assumption is not restrictive in practice due to the rather large size of mix banks and the relatively small size of sequences to be optimized, when applied as in our limited lookahead approach of Section 5.3.6.

We distinguish two different formulations for the generation of the assembly sequence. The AP formulation assigns all workpieces of the OEM sequence (1-10 in the example of figure 5.1) to a number of slots in the assembly line sequence. The TSP formulation, on the other hand, regards the assembly sequence as a path (in the example from 8 to 4) during which all workpieces have to be visited once and only within one tour. The sorting of finished workpieces on the mix banks (see the middle part of Figure 5.1) is modeled by a VRP. As in the VRP, we consider that all workpieces have to be assigned to specific mix banks once (representing the trucks of the VRP) and in a specific order inside each mix bank (the route). After sorting, the original sequence of the OEM is restored (downstream sequence) by withdrawing workpieces FIFO from the mix banks (see the bottom part of Figure 5.1). The logic to ensure this withdrawal is considered during the sorting stage as workpieces are only allowed to be placed in ascending indices on the mix banks.

5.3.2. Resequencing and Sorting

We next define the core of our mathematical model, which concerns the resequencing and sorting of the workpieces on the assembly line. We introduce e_{ki} as a binary variable indicating whether workpiece i of the original sequence is assembled at position k in the new assembly sequence. For the sorting, we use binary variable z_{ij} to determine whether workpiece j is placed directly behind i on the mix banks. u_{ij} is defined as a continuous variable that determines the number of workpieces to follow behind i on the mix bank, conditional to j being directly stored behind i on the same mix bank. For z_{ij} and u_{ij}, variables are only defined for $i, j = 0, \ldots, n, i < j$ to ensure that workpieces can only be placed in ascending order of indices on the mix banks.

We formulate the following constraints.

$$\sum_{k=1}^{n} e_{ki} = 1 \qquad i = 1, \ldots, n \tag{5.1}$$

$$\sum_{i=1}^{n} e_{ki} = 1 \qquad k = 1, \ldots, n \tag{5.2}$$

$$\sum_{i=0}^{j-1} z_{ij} = 1 \qquad j = 1, \ldots, n \tag{5.3}$$

$$\sum_{j=i+1}^{n} z_{ij} \leq 1 \qquad i = 1, \ldots, n-1 \tag{5.4}$$

$$\sum_{j=1}^{n} z_{0j} \leq m \tag{5.5}$$

$$\sum_{j=1}^{n} z_{0j} \geq \left\lceil \frac{n}{f} \right\rceil \tag{5.6}$$

$$\sum_{i=0}^{j-1} u_{ij} - \sum_{i=j+1}^{n} u_{ji} = 1 \qquad j = 1, \ldots, n \tag{5.7}$$

$$z_{ij} \leq u_{ij} \leq f z_{ij} \qquad i = 0, \ldots, n-1, j = i+1, \ldots, n \tag{5.8}$$

$$\sum_{j=1}^{n} u_{0j} = n \tag{5.9}$$

$$\sum_{k=1}^{n} k e_{ki} \leq \sum_{k=1}^{n} k e_{kj} + n\left(1 - z_{ij}\right) \qquad i = 1, \ldots, n-1, j = i+1, \ldots, n \tag{5.10}$$

$$z_{ij} \in \{0, 1\} \qquad i, j = 0, \ldots, n; i < j \tag{5.11}$$

$$e_{ki} \in \{0, 1\} \qquad k, i = 1, \ldots, n \tag{5.12}$$

The first two constraints resemble the assignment problem, where (5.1) ensures that each workpiece is assigned to one position in the assembly sequence and (5.2) that each position gets only one workpiece assigned. Constraints (5.3)–(5.9) represent the two-index single-comodity flow formulation of the VRP of Gavish and Graves (1978). (5.3) maintains that each workpiece on a mix bank has to be placed behind another workpiece with a lower index (or is the first, i.e., placed behind the dummy workpiece 0). (5.4) enforces that after a workpiece (or dummy workpiece) on a mix bank, not more than one workpiece can be placed directly behind and again only with an increased index. Due to increasing index values on the mix banks, we ensure the FIFO sorting principle. A mix bank will only be used when it has an item on the first position (i.e., after dummy product 0, $z_{0j} = 1$). Thus, by the sum $\sum_{j=1}^{n} z_{0j}$ in (5.5) we get the total number of mix banks used, which is limited to the number of mix banks available m. A valid inequality for improving the LP relaxation is given in (5.6), which places the lower bound on the minimum number of mix banks needed and is used in general VRP formulations (Toth & Vigo, 2014). (5.7) ensures that the value for u_{ij} decreases by one for each workpiece along the mix bank. This is done to let u_{ij} take the value of the number of workpieces to follow i in the mix bank. In (5.8), it is ensured that u_{ij} can only take positive values (bounded by the mix bank capacity f) if j has indeed been placed after i on the mix bank, i.e., when $z_{ij} = 1$. Note that u_{ij} is only defined when indices increase, i.e., we again ensure that the FIFO sorting principle is applied. Constraint (5.9) assures that the total number of workpieces on the mix banks placed behind the dummy workpiece 0 is equal to the sequence length n. The combination of the assignment problem and the vehicle routing problem is done in (5.10). This constraint establishes that, if j is directly placed behind i on a mix bank, i has to be produced before j during assembly. (5.11) and (5.12) restrict z_{ij} and e_{ki} to be binary.

Note that the first part of the problem ((5.1)–(5.2)) regarding the position of workpieces in the new assembly sequence can also be formulated as a traveling salesman instead of an allocation problem as above. This makes sense if sequence dependent factors are included in the problem (e.g., sequence dependent setup times). In this case, (5.1), (5.2), (5.10), (5.12) and the decision variables e_{kj} have to be omitted. We introduce x_{ij} as the binary variable indicating whether workpiece j directly follows workpiece i in the assembly sequence and decision variable y_{ij} as the number of workpieces to follow i in the assembly sequence, given that j is produced directly after i during assembly. We then define the following additional constraints.

$$\sum_{i=0}^{n} x_{ij} = 1 \quad j = 0, \ldots, n, j \neq i \tag{5.13}$$

$$\sum_{j=0}^{n} x_{ij} = 1 \quad i = 0, \ldots, n, j \neq i \tag{5.14}$$

$$\sum_{i=0}^{n} y_{ij} - \sum_{i=0}^{n} y_{ji} = 1 \quad j = 1, \ldots, n \tag{5.15}$$

$$0 \leq y_{ij} \leq n x_{ij} \quad i, j = 0, \ldots, n, i \neq j \tag{5.16}$$

$$\sum_{j=1}^{n} y_{0j} = n \tag{5.17}$$

$$\sum_{k=0}^{n} y_{ki} \geq \sum_{k=0}^{n} y_{kj} + 1 - n(1 - z_{ij}) \quad i = 1, \ldots, n-1, j = i+1, \ldots, n \tag{5.18}$$

$$x_{ij} \in \{0, 1\} \quad i, j = 0, \ldots, n, i \neq j \tag{5.19}$$

(5.13) ensures that all workpieces in the assembly sequence have one predecessor (where 0 is the start/end dummy workpiece). Likewise, in (5.14), it is ensured that exactly one successor is produced after each workpiece. (5.15) forces the values for y_{ij} to decrease by one from one workpiece to the next. The upper bound of y_{ij} is given by (5.16), as y_{ij} may only take positive values if j directly follows i, i.e., if $x_{ij} = 1$. (5.17) ensures that all workpieces are considered in only one route. (5.18) achieves the linking between the assembly and the mix bank sequence. If j is placed directly behind i in a mix bank (i.e., $z_{ij} = 1$), the number of workpieces that follow j in the assembly sequence ($\sum_{k=0}^{n} y_{kj}$) has to be smaller than the number of workpieces that follow i ($\sum_{k=0}^{n} y_{kj}$), i.e., j has to be produced after i. Finally, (5.19) is the binary condition of the decision variable x_{ij}.

5.3.3. Load Balancing Objective (LB)

We adapt the formulation of the load balancing objective for mixed-model sequencing problems from Boysen et al. (2009b) to integrate with (5.1)–(5.12). We refer to this as model LB in the following.

We introduce some additional notation. P is the number of stations to consider, with $p = 1, \ldots, P$. l_p denotes the length of each station, b_{ip} the processing time of workpiece i at station p, and c the takt time. An additional decision variable s_{pi} denotes the position of the operator at station p when the production of workpiece i starts. Lastly, decision variable w_{pi} determines the overload of the operator at station p after workpiece i has been produced. This overload is determined by the time needed to complete the workpiece in excess of the station's right border, i.e., the required work that the worker cannot cover himself without floating into the next station. Furthermore,

$M := \max_{p=1,\dots,P} \sum_{i=1}^{n} b_{ip}$ is a sufficiently large number. We add the following objective and constraints to the model.

$$\text{Minimize} \quad \sum_{p=1}^{P} \sum_{i=1}^{n} w_{pi} \tag{5.20}$$

s.t.

$$(5.1) - (5.12)$$

$$\begin{aligned} s_{pj} \geq s_{pi} + b_{ip} - w_{pi} - c - M\left(2 - e_{ki} - e_{k+1,j}\right) \\ i = 1, \dots, n, j = 1, \dots, n, i \neq j, k = 1, \dots, n-1, p = 1, \dots, P \end{aligned} \tag{5.21}$$

$$s_{pi} + b_{ip} - w_{pi} \leq l_p \qquad i = 1, \dots, n, p = 1, \dots, P \tag{5.22}$$

$$s_{pi}, w_{pi} \geq 0 \qquad i = 1, \dots, n, p = 1, \dots, P \tag{5.23}$$

The objective (5.20) sums up all overload situations over all stations and workpieces. Constraint (5.21) assures that the position of the operator at a given station p at the production start of j is dependent on the position at the production start of i, the necessary process time b_{ip} and a potential overload w_{pi}. After the production of i, the position of the worker is reduced by the takt time c. This constraint is only binding if j is directly produced after i, i.e., $e_{ki} + e_{k+1,j} = 2$. As each station is bounded by a given length l_p, overload of w_{pi} might occur if the position of the operator before the production of i plus the process time b_{ip} exceeds this length, as considered in (5.22). Lastly (5.23) defines the non-negativity of the new decision variables.

For this objective, the AP formulation shows better performance than the TSP formulation. However, both could be used as only little change is necessary.

5.3.4. Material Leveling Objective (ML)

Again, the formulation of Boysen et al. (2009b) is the basis for our material leveling objective, which extends the resequencing and sorting constraints of (5.1)–(5.12). As discussed in Boysen et al. (2009b), a multitude of weighting functions for the deviations between the actual and a target material consumption exist and they are, in any case, an approximation of the inclined costs of late and early supplies. We, therefore, adjust the basic formulation and use the absolute deviation, rather than the quadratic deviation, in order to remain with a linear model. This results in a $[P, abs|]$ model in terms of the classification introduced by Boysen et al. (2009b). This model has to use the AP formulation in order to be able to quantify cumulated material consumption, which cannot be directly determined by the TSP formulation as the assembly sequence is not numbered and thus the absolute position of each workpiece in the assembly sequence is not known. We refer to this as model ML in the following.

In addition to the resequencing and sorting constraints, we introduce R as the number of materials to consider, with $r = 1, \dots, R$. Parameters a_{ir} represent the bill of material of workpiece i regarding material r. For a given material r, we determine the average consumption per workpiece $\sum_{i=1}^{n} a_{ir}/n$ and calculate the target cumulated consumption t_{kr}, at position k of the assembly sequence, by summing the average consumption over all workpieces assembled up to k: $t_{kr} = \frac{k}{n} \sum_{i=1}^{n} a_{ir}$. We furthermore need decision variables d_{kr} as the absolute deviation between the target and the actual cumulated consumption of material r when assembling the kth workpiece.

$$\text{Minimize} \quad \sum_{k=1}^{n} \sum_{r=1}^{R} d_{kr} \tag{5.24}$$

s.t.

$$(5.1) - (5.12)$$

$$d_{kr} \geq t_{kr} - \sum_{q=1}^{k} \sum_{i=1}^{n} e_{qi} a_{ir} \qquad k = 1, \dots, n, r = 1, \dots, R \tag{5.25}$$

$$d_{kr} \geq \sum_{q=1}^{k} \sum_{i=1}^{n} e_{qi} a_{ir} - t_{kr} \qquad k = 1, \ldots, n, r = 1, \ldots, R \tag{5.26}$$

The objective in (5.24) minimizes the absolute deviation between the actual and the target cumulative consumption of all materials and workpieces. Apart from the constraints of the core module, we introduce constraints (5.25) and (5.26) to determine the absolute difference between the actual and the target material consumption at the production of each workpiece, where (5.25) determines any negative and (5.26) any positive deviation. Note that, due to those constraints, a non-negativity constraint for d_{kr} is not necessary.

5.3.5. Color Batching Objective (CB)

In contrast to the aforementioned models, we use the TSP formulation for the resequencing problem for the color batching objective as in Spieckermann et al. (2004). The reason for this lies within the cost structure, which is dependent on the position of one workpiece in relation to the other workpieces in the assembly line, rather than the absolute position as in Section 5.3.4. We refer to this as model CB in the following.

The goal of this objective is to minimize the color changes during production, which are indicated by $c_{ij} = 1$, if the color of workpiece j differs from the color of workpiece i and 0 otherwise. We formulate the mathematical problem as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{5.27}$$

s.t.

$$(5.3) - (5.9), (5.11), (5.13) - (5.19)$$

The only new part in this formulation is the objective (5.27), which sums up all color changes in the assembly sequence.

5.3.6. Limited Lookahead Approach (LL)

To facilitate faster computation and lower model complexity, we apply a limited lookahead approach (for a formal definition of lookahead see Dunke & Nickel, 2016) to each of the models mentioned. To do so, we iterate over a subset of ν workpieces to run the MILP model as described above. We then fix the first κ decisions of the result and move forward. We thus have $I = \lceil n/\kappa \rceil$ iterations. In each iteration $\jmath$, we include the first $\bar{n} = \min\{\kappa\,(\jmath - 1) + \nu, n\}$ workpieces, while the position of the first $\kappa \cdot (\jmath - 1)$ workpieces on the assembly line are already fixed based on the decisions of previous iterations. We solve the reduced model and then fix all decisions up to $\bar{k} = \jmath \cdot \kappa$. We then shift the lookahead window further ahead by κ workpieces and start the next iteration.

As an illustrative example, see Figure 5.2. The solution is build step by step with each iteration $\jmath$ for an instance with $n = 15, \nu = 8, \kappa = 5$. In the first iteration, the model is solved for workpieces 1 to 8 (in gray). In the next iteration, the first $\kappa = 5$ workpieces of that solution are fixed (in black) and a new solution for the next $\nu = 8$ workpieces from 6 to 13 are generated, from which again the first 5 workpieces are fixed in the next iteration, and so on. For a more detailed explanation of the algorithm, see Appendix 5.A.

Figure 5.2.: Illustration of the limited lookahead heuristic with $\nu = 8$ and $\kappa = 5$

		Work pieces														
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	1	8	5	6	1	3	4	7	2							
$\jmath$	**2**	8	5	6	1	3	10	2	13	11	7	12	4	9		
	3	8	5	6	1	3	10	2	13	11	7	12	15	4	14	9

■ Fixed by previous iteration(s) ■ Workpieces in optimization ($\underline{n}$ to $\bar{n}$)

We refer to this as model LL and to the full MILP models as FM in the following.

5.4. Numerical Study

5.4.1. Data Generation and Setup

Values for the processing times under the load balancing objective have been created by a procedure adapted from Dörmer et al. (2015). We distinguish four different product classes (basic, standard, premium and luxury). For each station p and each product class cl, we randomly generate four processing times ($pt^1_{p,cl}$, $pt^2_{p,cl}$, $pt^3_{p,cl}$, $pt^4_{p,cl}$) based on the intervals in Table 5.1. We repeat the generation until the following condition is met $pt^1_{p,cl} < pt^2_{p,cl} < pt^3_{p,cl} < pt^4_{p,cl}$.

Table 5.1.: Intervals of processing times (Dörmer et al., 2015)

Processing times	Basic	Standard	Premium	Luxury
$pt^1_{p,cl}, pt^2_{p,cl}$	$[0,1]$	$[0,1]$	$[0,1]$	$[0,1]$
$pt^3_{p,cl}, pt^4_{p,cl}$	$[1,1.5]$	$[1,2]$	$[1,3]$	$[1,5]$

We next assign probabilities that a certain processing time value is picked to each product class. This is needed later when we create individual processing times for each workpiece in the sequence. Furthermore, each product class receives a probability that a workpiece in the sequence is assigned to this class. The corresponding values can be found in Table 5.2.

Table 5.2.: Probabilities of processing times and of demand per product class (Dörmer et al., 2015)

Product class (cl)	$Pr(pt^1_{p,cl})$	$Pr(pt^2_{p,cl})$	$Pr(pt^3_{p,cl})$	$Pr(pt^4_{p,cl})$	Demand prob.
Basic	50 %	25 %	25 %	-	33 %
Standard	25 %	50 %	25 %	-	33 %
Premium	-	25 %	50 %	25 %	17 %
Luxury	-	25 %	25 %	50 %	17 %

Generating a workpiece i for the sequence starts by creating a uniform random value in the interval $[0,1]$. Depending on this value, we assign i to one of the product classes based on the demand probabilities of Table 5.2. Later, in our extended dataset, these probabilities will be changed. We assign i to product classes by assuming a cumulated distribution given by these values, i.e., random values in the interval $[0,0.33)$ generate a basic workpiece, in the interval $[0.33,0.66)$ generate a standard workpiece, etc.. As we do this random assignment to product classes for each workpiece individually, we thereby create a first randomized sequence in terms of product classes. Next, we proceed for each workpiece i and station p by randomly assigning the processing times ($pt^1_{p,cl} - pt^4_{p,cl}$) based on the processing time probabilities of Table 5.2 for the relevant product class.

Doing so, we achieve that, even if we have two workpieces of the same product class, they will have different processing times in each station.

After generating the sequence and the process times for each workpiece and station, we rescale all process times to achieve an average of 0.95 time units in each station. As the cycle time c is fixed to 1, we thereby get a 95 % utilization. Also, this utilization factor is varied within our extended dataset in the numerical study. Afterwards, the length of each station is set to $l_p = \lceil pt^4_{p,Luxury} \rceil$.

For obtaining the bill of materials in the material leveling objective, we proceed likewise, however considering different materials r instead of stations, and multiplying the intervals of Table 5.2 with a factor of 10. Any randomly generated material requirement is rounded up to the next integer value. We additionally consider the possibility that a certain workpiece does not need the material at all (i.e., $a_{ir} = 0$). We randomly set a_{ir} to zero, based on a 20 % probability throughout all experiments.

The color assignment for the color batching objective is done uniformly random for each workpiece in the sequence on the basis of C different colors. The calculation of c_{ij} is then done according to the definition in Section 5.3.5.

For the experiment, we consider two different datasets: a basic dataset that is used for comparing the full model with the limited lookahead heuristic and an extended dataset that is used for giving insights into instances of practical size and that analyzes the impact of more variation in the parameters based on the limited lookahead heuristic. See Table 5.3 for the details.

The setup was inspired by similar publications and our projects with the automotive supplier. As we are in a JIS setting, the work-in-process and thus the scope for any resequencing is bounded by the short time between the arrival of a signal regarding the final sequence from the OEM and the actual shipment (Swaminathan & Nitsch, 2007), which is only a few hours (Meyr, 2004). It can be expected that the scope is further reduced by the number of items that fit in a truck as especially bulky items are more likely to be produced JIS by the supplier (Fredriksson, 2006; Swaminathan & Nitsch, 2007). This is why we focused on a setting with up to 40 workpieces for our comparison between the model formulations in the basic dataset, which represents the order size for the car seat supplier in Doran (2001). In the extended dataset, we compare larger instances with up to 100 workpieces and 50 stations/products/colors. Similar sizes of problem instances can also be found in other resequencing publications:

Table 5.3.: Numerical study setup

Dataset	**Basic**	**Extended**
n	$5, 10, \dots, 40$	$40, 60, 80, 100$
$P/R/C$	$1, 2, 3, 4, 5$	$10, 30, 50$
m	$2, 3, 4, 5$	$2, 5, 10, 15$
f	$\lceil n/m \rceil$, $\lceil n/m \rceil + 1$, $\lceil n/m \rceil + 2$	$\lceil n/m \rceil$, $\lceil n/m \rceil + 1$, $\lceil n/m \rceil + 2$
Demand probabilities	$[0.33, 0.33, 0.17, 0.17]$	$[0.33, 0.33, 0.17, 0.17]$, $[0.25, 0.25, 0.25, 0.25]$, $[0.4, 0.4, 0.1, 0.1]$
Utilization (LB and ML)	0.95	$0.85, 0.9, 0.95$
Number of replications	10	10
Model formulations	FM, LL (all parameter variations), No-FIFO	LL (selected parameter setting)

up to 20 colors/stations and 100 workpieces in Boysen et al. (2010, 2011); Boysen and Zenker (2013); Ding and Sun (2004); Han and Zhou (2010); Lahmar et al. (2003); Lin et al. (2011); Spieckermann et al. (2004); Sun et al. (2015). We chose ten replications per instance in order to be consistent with Boysen et al. (2010); Boysen and Zenker (2013); Gujjula and Günther (2009); Sun et al. (2015).

We assume that assembly starts with the first workpiece and no task has been done before, i.e., workers are starting from their initial position, no material has been previously demanded and no changeover cost occurs for the first workpiece. However, this could be adjusted easily in the respective model input parameters.

We implement the models in Fico Xpress 7.9 and run experiments on a 12-core Intel Xeon CPU with 2.6 GHz and 32 GB RAM. For FM we first solve each instance with the restriction that the OEM sequence has to be produced as-is. We store this solution and load it into the original model to achieve a warm start with a feasible solution. Each instance is stopped after the optimal solution has been found or 1,000 seconds have elapsed. We restrict the models to this time limit as longer computation times are infeasible for the operational setting in practice. For a fair comparison, each iteration of the limited look-ahead model is restricted to $1,000/I$ seconds to ensure a maximum calculation time of 1,000 seconds over all iterations. To compare solutions we measure the objective value and gap of the MILP models. Additionally, we calculate the objective values of the original sequence and the sequence that can be achieved without the restriction on FIFO sorting via mix banks as lower and upper bounds. We use the term No-FIFO to represent this variant. We model the No-FIFO approaches by taking the model formulations of Sections 5.3.2–5.3.5 and eliminate the decision variables z_{ij} and u_{ij} and constraints associated with the storage on the mix banks (5.3)–(5.11) and (5.18). Our main performance indicator when comparing the models is the savings in the objective function achieved against the lower bound objective value of the original sequence. In addition, we capture complexity by determining the distance between the original sequence and the best supplier sequence under the No-FIFO environment. We do so by calculating the average one-dimensional distance between positions in the sequence of each workpiece, i.e., $\bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n} e_{ki} \cdot |i-k|$, where e_{ki} is defined as in Section 5.3.2.

For instances where LL rendered a worse sequence than the original sequence, due to its heuristic nature, we assume that the original sequence is taken and thus the savings are zero.

5.4.2. Results of Basic Dataset

5.4.2.1. Computation Times

Before a more thorough analysis of the results of our numerical study, we make a sensitivity analysis on the parameters for the limited look-ahead model, i.e., for the length of the look-ahead horizon ν and the number of workpieces to be fixed in each iteration κ. We apply the full numerical design above on four different setups, νxκ = 8x3, 8x5, 10x3 and 10x5. We choose the values for ν in such a way that they represent sequence lengths which are quickly solvable even in the full model (see also Table 5.5). The values for κ have been set to be smaller than ν to reflect the rolling horizon effect and they have been varied to show the different effects they have on solution time and savings.

We observe in Table 5.4 that average and maximum solution times are minimized by the 8x5 parameter setup. However, for all objectives, better solutions are found by the 10x3 setup. Both results are intuitive, as the 8x5 setup considers the smallest sequence length per iteration (8) and the smallest number of iterations ($\lceil n/5 \rceil$). The 10x3 setup, in

Table 5.4.: Solution times (in seconds) and savings of different limited look-ahead parameters

	Parameters ν x κ	Time (avg)	Time (max)	Savings (avg)
Load Balancing	8x3	43.44	420.34	52.96 %
	8x5	**12.99**	**246.56**	50.46 %
	10x3	92.67	679.98	**60.79 %**
	10x5	114.59	1000.00	60.59 %
Material Leveling	8x3	3.01	12.95	54.32 %
	8x5	**1.36**	**7.36**	52.44 %
	10x3	5.63	28.80	**56.14 %**
	10x5	3.89	19.97	55.59 %
Color Batching	8x3	5.67	51.33	61.17%
	8x5	**2.70**	**26.33**	56.97 %
	10x3	17.60	286.77	**64.96 %**
	10x5	10.42	248.71	63.88 %

turn, has the longest sequence length per iteration and the highest number of iterations, which means that it increases the optimization potential. As we are in an operational setting, we apply the 8x5 parameter setup in all subsequent analyses, thus focusing on solution time rather than optimality. This also ensures a more conservative comparison, as time is also a crucial component in a practical application. For a detailed look at the savings realized by the fastest and best parameter combination in different problem sizes, see Figures 5.4a-5.4c in Section 5.4.2.3.

From a managerial perspective, this analysis shows clearly that, prior to fixing the model parameters in practice, one should conduct a thorough ex-post analysis of the solution time and savings that would have been realized in past sequences. Based on this analysis, the most appropriate setting can be chosen, reflecting the respective objective, solution time restriction, and solution quality.

Comparing the average and maximum solution times between FM and LL over all instances of our numerical study (Table 5.5), further indicates the applicability of the limited look-ahead approach to practical large scale settings. In comparison, the full MILP model exponentially increases in computation time.

Table 5.5.: Solution time of FM and LL (in seconds)

	Load Balancing				Material Leveling				Color Batching			
	FM		LL		FM		LL		FM		LL	
n	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max
5	1	1	1	1	1	1	1	1	1	1	1	1
10	1	8	1	3	2	13	1	1	1	14	1	2
15	46	1,000	2	33	373	1,000	1	2	268	1,000	1	3
20	222	1,000	6	44	978	1,000	1	4	859	1,000	2	4
25	462	1,000	12	104	1,000	1,000	2	3	998	1,000	3	8
30	658	1,000	17	136	1,000	1,000	2	4	1,000	1,000	4	9
35	841	1,000	29	176	1,000	1,000	2	7	1,000	1,000	5	11
40	935	1,000	38	247	1,000	1,000	3	5	1,000	1,000	7	26
Avg.	*395.47*		*12.99*		*669.28*		*1.38*		*640.91*		*2.69*	

5.4.2.2. Solution Quality

The overall savings in comparison to the initial sequence are significant under all three objectives, with savings of between 30 - 57 % on average depending on the objective and the chosen model (Table 5.6).

Table 5.6.: Average Savings (in %) and distance $\bar{\delta}$

	Load balancing			Material Leveling			Color Batching		
n	$\bar{\delta}$	FM	LL	$\bar{\delta}$	FM	LL	$\bar{\delta}$	FM	LL
5	1.72	10.00	10.00	1.58	28.82	28.82	1.32	32.50	32.50
10	3.61	52.99	50.00	3.39	47.74	46.17	2.70	62.91	52.10
15	4.93	73.40	63.08	5.15	55.82	52.10	4.31	72.53	58.75
20	6.96	73.00	54.73	6.80	57.23	54.15	5.74	63.62	61.36
25	8.25	74.37	66.72	8.11	38.84	56.67	6.93	46.78	61.90
30	9.55	52.60	56.16	9.68	11.95	60.21	8.24	19.29	62.15
35	11.34	32.79	54.62	11.01	0.73	60.38	9.98	8.00	63.13
40	13.53	17.60	48.35	12.89	0.00	60.98	11.83	2.32	63.63
Avg.	*7.49*	*48.35*	*50.46*	*7.32*	*30.14*	*52.44*	*6.38*	*38.49*	*56.94*

Although differences between the limited lookahead approach (LL) and full model (FM) approach are small on average for the load balancing objective (2 %-points), differences increase significantly for the material leveling (22 %-points) and color batching problems (19 %-points), all in favor of LL. The significant differences for the latter two stem from the increased complexity, which goes in line with the exponentially increasing solution times as observed in Table 5.5. Another takeaway from Table 5.6 is that the average distances $\bar{\delta}$, quite naturally, rise with sequence length, yet are still quite similar in all three objectives.

Next, we observe the quality of the limited lookahead heuristic specifically for those instances where the full MILP was able to find the optimal solution. The *GapLL* value is defined as the relative increase in the objective function between the limited lookahead heuristic and the full MILP. We define *ShareOpt* as the share of instances, where FM found the optimal solution. Within those instances, *ShareLL* defines the share of solutions of the heuristic with a *GapLL* smaller than 1 % to the optimal FM solution. Another value *SGapLL* is defined as the difference in savings (in %-points) between the full MILP and the limited lookahead heuristic. We compare the average values for *ShareLL*, *GapLL* and *SGapLL* in Table 5.7.

Table 5.7.: Comparison of LL to FM for optimal solutions

	Load Balancing		Material Leveling		Color Batching	
	8x5	10x3	8x5	10x3	8x5	10x3
ShareOpt (in %)	65.08		35.77		37.66	
ShareLL (in %)	66.20	80.22	56.20	81.42	47.79	81.81
GapLL (in %)	39.32	26.31	3.94	0.94	39.16	9.74
SGapLL (in %-points)	11.08	4.25	1.73	0.41	9.38	2.11

Note: The table considers all instances where FM found the optimal solution.

ShareLL reveals that the heuristic finds very good solutions in around 80 % of the cases with the 10x3 setup and in between 47 and 66 % with the 8x5 setup. The GapLL values show that, for the most conservative setup of the LL heuristic (8x5), the objective value increases by 39 % for the Load Balancing and Color Batching objective, while in the Material Leveling Objective the gap is less than 4 % due to the heuristic. When using the 10x3 setup for LL, those values drop considerably to 26.31 % for Load Balancing, 0.94 % for Material Leveling and 9.74 %-points in the Color Batching objective.

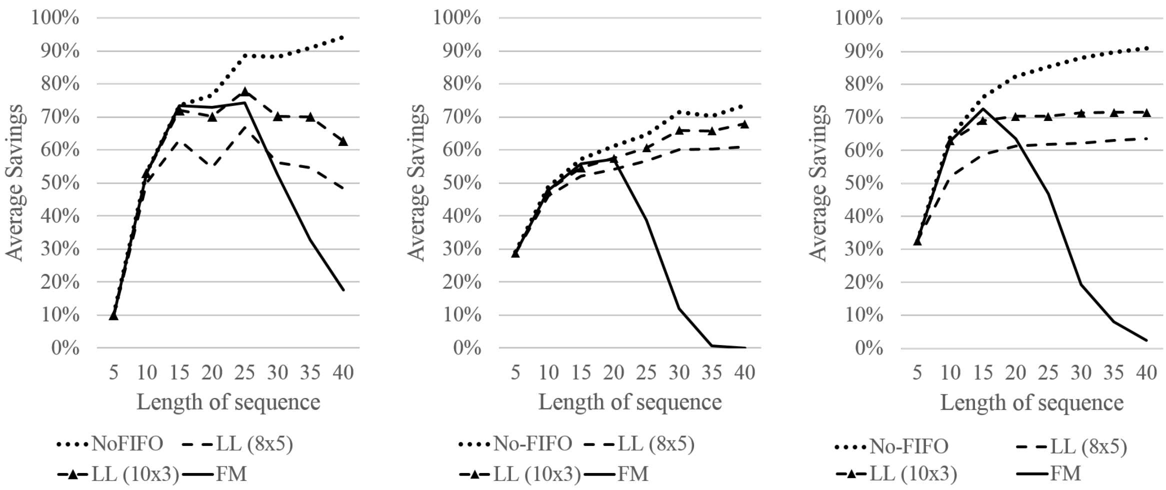
We observe how the gap to optimality translates into actual lost savings in the values for SGapLL. The results show that the impact on the relative savings is far smaller than GapLL indicates. The reason is in the objective values, which might be quite small for good or optimal solutions in comparison to the original sequence.

While this shows that using the heuristic comes with costs in terms of gaps to the optimal solution, it also shows that varying the setup of the heuristic might counteract those costs to some extent.

5.4.2.3. Impact of FIFO Restriction

As we require that workpieces have to be retrieved FIFO from the mix banks, we next investigate how much is lost by this constraint in comparison to a No-FIFO approach, where the sequence can be altered without any further constraint, thus constituting a pure sequencing problem.

Figure 5.3.: Average savings of full MILP, Limited Lookahead and No-FIFO



(a) Load Balancing (b) Material leveling (c) Color Batching

We see in Figures 5.4a-5.4c that, with growing problem size, the limited look-ahead approach performs much better than the full MILP that covers the whole sequence at once. For the load balancing objective, the average loss in savings against the No-FIFO model amounts to 21.46 and 11.12 %-points for the limited look-ahead setups as opposed to 23.57 %-points for the full MILP model. However, up to a sequence length of 10, no differences between the objective values of the full MILP and No-FIFO approach have been observed.

For the material leveling objective, the loss of the FIFO restriction varies considerably: between solution approaches with 29.40 %-points for the full MILP and 7.10 and 3.40 %-points for the limited look-ahead setups. Also, even for the smallest number of mix banks, a difference between the solutions to No-FIFO can be observed.

Likewise, observations can be made for the color batching objective, however, the difference between these and the No-FIFO solution is much larger. The average loss in savings against the No-FIFO model amounts to 11.18 and 19.20 %-points for the limited look-ahead setups as opposed to 37.64 %-points for the full MILP model. The full MILP model can obtain the same solutions as the No-FIFO approach for all instances with the smallest sequence length of 5. For some of the instances with a sequence length of up to 25, the model still finds solutions as good as those found by the No-FIFO approach.

To further examine the impact of the sequence length, we show the average absolute savings per workpiece in Table 5.8.

Table 5.8.: Average absolute savings per workpiece for LL (8x5)

n	LB	ML	CB
5	0.005	4.92	0.20
10	0.065	13.90	0.33
15	0.078	18.43	0.39
20	0.085	20.95	0.41
25	0.088	22.51	0.40
30	0.077	29.85	0.41
35	0.089	29.33	0.42
40	0.080	34.07	0.42

For all three objectives, we see the impact of the sequence length on the optimization potential. Until approximately 15 workpieces, savings per workpiece increase for all objectives. However, for the load balancing objective, savings per workpiece seem to be unstable and they even drop towards larger sequence lengths. For the material leveling approach, however, savings per workpiece seem to steadily increase and for the color batching approach, savings seem to remain stable after a certain sequence length. Hence, managers should take into account that too small sequences are not very likely to produce large optimization potential. Large sequences, on the other hand, increase the (to some degree limited) optimization potential, yet they reduce the computational performance.

5.4.2.4. Mix Bank Design

One important question that remains to be answered is the design of such a system in practice. We investigate the impact of the number of mix banks m and capacity per mix bank f by calculating the cumulative absolute savings when increasing the respective parameter.

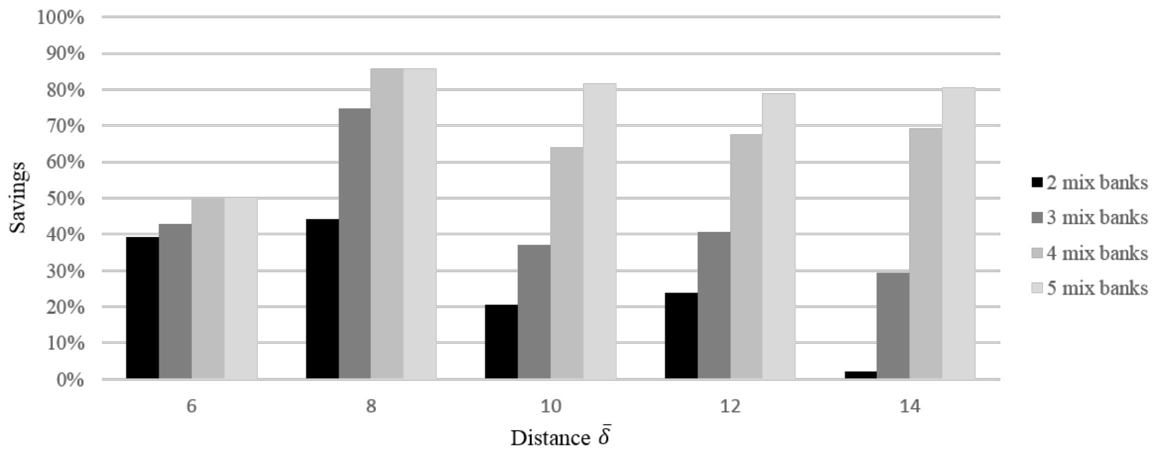
Table 5.9.: %-point increase in cumulative savings by number of mix banks and capacity

	m	$f = \lceil n/m \rceil$	$f = \lceil n/m \rceil + 1$	$f = \lceil n/m \rceil + 2$
Load Balancing	2	+ 35.42	- 0.77	+ 0.73
	3	+ 8.43	+ 1.14	+ 2.09
	4	+ 7.59	+ 2.32	+ 0.92
	5	+ 6.00	+ 2.90	+ 0.80
Material Leveling	2	+ 23.95	+ 0.95	+ 0.05
	3	+ 1.02	+ 1.01	+ 0.75
	4	+ 6.19	+ 1.15	+ 0.47
	5	+ 4.23	+ 3.47	+ 0.79
Color Batching	2	+ 32.68	+ 0.60	- 0.40
	3	+ 3.29	+ 1.48	- 0.07
	4	+ 3.11	+ 1.41	+ 1.06
	5	+ 1.85	+ 3.73	+ 0.90

In Table 5.9, we focus on solutions of FM. One can clearly observe that two mix banks provide the largest improvement, as expected. However, (smaller) gains can still be obtained with additional mix banks. These findings are in-line with similar investigations of Boysen and Zenker (2013). On the other hand, an increase in the capacity per mix bank f has only little effect most of the times. Increasing the mix bank parameters can have a negative impact, as adding capacity slack induces more computational complexity. However, one should keep in mind that the solution quality of FM deteriorates for large problem sizes, which is why the results should be interpreted with care.

When analyzing the effects that trigger a high benefit of an additional mix bank, we can observe that, apart from the length of the sequence (as we observed in Section 5.4.2.3), the distance between the original and the best supplier sequence under the No-FIFO environment has a significant impact (see Figure 5.5).

Figure 5.5.: Impact of $\bar{\delta}$ on the benefit of additional mix banks (Example for LB and $n = 30$)



This figure illustrates the sequence length $n = 30$ in the load balancing objective, although the same effects hold for other objectives and sequence lengths. We show the values for one specified sequence length in order to show that not only the sequence length itself but the distance $\bar{\delta}$, too, impacts the benefit. When following the increased values of $\bar{\delta}$ step by step, we observe that, on the one hand, the benefit of the second mix bank decreases in tendency, and, on the other hand, the spread in savings between a smaller and a larger number of mix banks widens. We can conclude that, while taking the sequence length into account is important, the quality of the OEM sequence, measured by the distance to the best supplier sequence under the No-FIFO environment, also has to be taken into account when designing the system in the long run. A possibility to do so would be to perform a back testing of different setups for the mix banks over historically obtained OEM sequences. Nevertheless, it might be difficult to obtain a reliable prediction of the distance for the future based on these historical values.

Hence, managers should be made aware of the possibility that significant consequences in the realizable savings might occur with a certain setup of mix banks whenever OEM sequences systematically change in the future.

5.4.3. Results of Extended Dataset

We next investigate the extended dataset to observe the effects of large-scale problems and different settings for demand patterns and utilization. In Table 5.10, we show the average savings and solution times in the dimensions of n, $P/R/C$ and m.

Table 5.10.: Average savings and solution times for extended dataset

	Load Balancing		Material leveling		Color batching	
n	Savings	Time	Savings	Time	Savings	Time
40	48.05 %	342.45	31.70 %	7.93	19.29 %	3.06
60	36.78 %	823.43	30.54 %	14.37	19.21 %	12.86
80	17.52 %	792.34	28.55 %	27.15	19.42 %	32.96
100	11.27 %	702.02	28.61 %	51.08	19.15 %	75.43
$P/R/C$	Savings	Time	Savings	Time	Savings	Time
10	46.09 %	359.24	39.76 %	10.63	35.09 %	35.11
30	24.08 %	685.46	26.88 %	24.93	14.29 %	30.18
50	15.04 %	950.48	22.91 %	39.83	8.43 %	27.94
m	Savings	Time	Savings	Time	Savings	Time
2	25.17 %	542.23	22.18 %	24.12	18.65 %	7.13
5	29.45 %	724.30	31.96 %	25.45	19.60 %	44.64
10	29.60 %	700.72	32.60 %	25.61	19.41 %	43.93
15	29.40 %	693.00	32.65 %	25.35	19.41 %	28.60

The impact of n on the results differs from objective to objective. While the color batching objective seems to result in a stable solution quality, the savings in the material leveling objective decline and plummet for the load balancing objective. In Section 5.4.2.3, we found the same effects for the load balancing and color batching objective. However, the material leveling objective showed slightly rising savings for the smaller sequence lengths we investigated there. The effect on the load balancing objective becomes clear when we observe the solution times. While average solution times for material leveling and color batching are still far below the 1,000 seconds cut-off, the load balancing objective results in far longer solution times. Thus, there are instances where no integer solution is found within the time limit and then zero savings are achieved. However, in all three objectives, solution times increase linearly rather than exponentially.

For the effect of larger $P/R/C$ in all objectives on savings, we see a similar development. With rising values, the solution quality diminishes. This is due to the additional complexity for the solver to find good solutions, caused by a rise in a number of stations, materials or colors. As for the solution times, we observe two different phenomena: it rises steadily for the load balancing and material leveling, whereas there is a decrease for the color batching objective. This behavior can be explained by the fact that the number of decision variables increases with the number of stations and materials for the first two objectives, whereas the number of decision variables remains the same for the color batching problem.

In the last dimension, we observe the effect of m. Interestingly, m has a rather small influence on the average solution time for the load balancing and material leveling objective. Average savings significantly increase if we use between 2 and 5 mix banks and remain stable for a larger number of mix banks in those two objectives. This goes in hand with our findings of Subsection 5.4.2.4. The effects for the color batching objective differ to some degree. While solution times rise sharply from two to five mix banks, average savings only increase slightly.

Table 5.11.: Impact of utilization and demand pattern on average savings

		Demand pattern		
	Utilization	More Standard	Basic	More Premium
	0.85	32.02 %	33.28 %	35.18 %
LB	0.9	26.66 %	28.66 %	30.88 %
	0.95	20.85 %	23.47 %	24.61 %
	Average	*26.51 %*	*28.47 %*	*30.23 %*
ML	-	28.98 %	30.06 %	30.50 %

Table 5.11 shows the effects of varying utilization and demand patterns. We first observe the impact of varying demand patterns for the load balancing and material

leveling objectives. Note that the ML objective does not have a utilization, which is why we compare the effects on the overall averages. We can see that, for the load balancing objective, a variation of the demand pattern has a larger impact on the savings than in the material leveling objective. Savings increase in LB when the premium products have a larger share. This is due to the resulting larger variability in the process times within a sequence and the thus higher savings potential. The effects for ML are marginal and not strictly increasing with a larger share of premium products.

Regardless of the demand pattern in the LB objective, a smaller utilization of stations leads to far better solutions. This is due to the higher slack capacities on the stations and thus more solution possibilities for the optimization.

What managers should take away from this subsection is that large scale problems can be successfully solved for the material leveling and color batching problem by the limited lookahead heuristic, but this should be considered with care for the load balancing objective. For very large load balancing problems, managers may consider the option of a longer algorithm run time or else of only using a subset of the most critical stations of the assembly line to reduce computational complexity. Furthermore, a very large number of mix banks does not really provide benefits in solution quality after a certain point (see also our findings from Subsection 5.4.2.4). The approach is most beneficial if the number of products, materials and colors is not too high for any objective and the utilization of stations is low and more premium products are demanded within the load balancing objective.

5.5. Conclusion

We provided a MILP formulation for the resequencing problem under the constraint that restoration of the original sequence can be achieved by FIFO extraction from mix banks in order to fulfill just-in-sequence demand. Due to the adaptive formulation of the approach, we implemented three different objectives and provided insights into the solution quality and computation times in numerical studies answering our four research questions of Section 5.1.

- (RQ1): The numerical study showed that significant savings, > 50 % on average, are achieved in all objectives for a basic dataset, while only small sequence lengths can be solved to optimality within a decent time by the MILP.

- (RQ2): Based on our numerical results, we were also able to generate knowledge about the design of such a system. We conclude that the number of mix banks should be decided upon after thorough analysis of historical OEM sequences, as not only the length but also the quality of the OEM sequence drives the saving potential substantially. However, on average and as long as the distance between the supplier optimal sequence and the OEM sequence is not too large, having 2 mix banks already generates the largest impact, while additional mix banks bring some extra saving potential.

  From a managerial perspective, these effects of the number of mix banks and their capacity on the operational performance might serve as an input for a more detailed analysis of the investment decision for mix banks. While higher savings in the three investigated objectives are favorable, one has to bear in mind that implementing mix banks also comes with certain costs.

  One aspect of those costs is associated with the implementation of, for instance, conveyors and floor space and with the handling of a mix bank (e.g., maintenance, withdrawal of workpieces). Nevertheless, mix banks are considered to be cost efficient (Spieckermann et al., 2004). While those costs can be estimated by means of investment and activity based costing approaches, it is much harder to associate costs with, e.g., the sum of deviations in the material leveling objective. If the three objectives cannot be directly represented in terms of cost, managers will have to make a trade-off decision between savings in the objective and costs for the mix banks, which are subjective to a company's individual operation setting. The above analysis can provide operational key performance indicators for the decision-maker in that respect.

- (RQ3): Our proposed limited look-ahead heuristic produces good results in short calculation times, even for large-scale instances. However, the parameters for the heuristic should be carefully analyzed based on historical sequences in order to find a suitable mix of solution quality and computation time under the applied objective.

- (RQ4): Based on the variations in the investigated operational settings of our controlled testbed, we conclude that the implementation of such a resequencing system is most beneficial if

- for any objective, the sequence length is not too short (approximately > 15 workpieces),
- the sequence length is not too long (around < 40 workpieces) for the load balancing objective,
- the number of products, stations and colors is small,
- more complex products, in terms of handling effort and material consumption, are demanded and
- the average utilization of stations in the load balancing objective is small.

As we pointed out earlier, actual cost savings depend on several factors of a company's operation. Nonetheless, we would like to provide the following calculation examples to give the reader an idea about the possible cost savings. These are based on the average results from our extended dataset for a sequence length of 40.

For the load balancing objective, let us assume that the takt time and all other values in the numerical study are in minutes. In the extended dataset, 25.83 minutes of overload are saved on average within the 40 minutes sequence. Scaled to a full working shift of 8 hours or $8 \cdot 60/40 = 12$ sequences, this means 310.01 minutes. Assuming that the overload situation relates to utility workers who work additionally on the workpiece, we consider 40 Euro per hour as a penalty. This would lead to cost saving of 206.68 Euro per shift. For a whole year, this relates to 45,468.75 Euro if we assume 220 working days.

For the material leveling approach, assume that reductions in the deviation lead to smaller inventories, as safety stocks can be reduced. The average absolute deviation in the original sequence in the extended study for $n = 40$ is 24,080.62 units. This value is, however, cumulated along the sequence. We thus correct the value by the average sequence position in which the deviation occurred in the sequence, which is simply assumed to be the half sequence length. Thus we get $24,080.62/20 = 1,204.03$ units deviation to the average consumption per sequence. Taking $1,204.03$ as the standard deviation around the mean, a service level of 99 % and a normally distributed demand, a safety stock of $k \cdot \sigma = 2.32 \cdot 1,204.03 = 2,800.99$ units is needed for every sequence. The same calculation holds for the LL heuristic, with average absolute deviations of $17,326.55$ units, leading to a safety stock of $2,015.38$ units on average. Assuming holding costs of 0.1 Euro per day and unit, we come to safety stock costs of $739,462.67$ Euro without and $532,059.97$ Euro with resequencing per year (assuming 220 days and one shift per day with 12 sequences). Hence, a cost saving of $207,402.71$ Euro per year is achieved.

In the color batching objective, 7.00 setups are saved on average for $n = 40$ in the extended dataset. We assume average changeover costs of 250 Euro, based on the assumptions of the uniform distributed changeover costs as in Lahmar and Benjaafar (2007). This leads to $250 \cdot 7 = 1,750$ Euro cost reduction per sequence and for 12 sequences a day and 220 days, to cost savings of 4.62 million Euro per year.

For future research, it might be interesting to consider sorting devices apart from mix banks, e.g., pull-off tables. While pull-off tables have received far more attention than mix banks (Boysen et al., 2012), our setting of resequencing under just-in-sequence demand has not been tackled so far. Pull-off tables allow more flexibility in the possible shifting of products than mix banks. They can move products to an arbitrary position in the back and towards the front of the sequence by n positions, with n being the number of pull-off tables. As opposed to this, the movement of workpieces in mix banks is constrained in both directions by the unoccupied capacity left on all mix banks combined. Nevertheless, literature reports that the number of pull-off tables in practice is rather small (Boysen et al., 2012), while the number and capacity of mix banks are rather large (Boysen & Zenker, 2013; Choi & Shin, 1997).

While the limited lookahead heuristic shows promising results, other solution approaches might be worth investigating to further reduce the gap to the optimal solution. Especially, given that the model formulation consists of a combination of well-known problem formulations such as VRP, TSP and AP, problem-specific heuristics (such as the savings heuristic for VRP and TSP) might be promising.

We investigated the resequencing approach under three different objectives, each of the objectives individually. In practice, several objectives should be considered at the same time for an overall efficient process (e.g., load balancing and color batching). Therefore, an extension to a multi-objective approach (as, e.g., in Chica et al., 2016) should be investigated in the future. This has also been identified as an area of further research in resequencing by Boysen et al. (2012).

5.A. Appendix: Limited Lookahead Algorithm

Algorithm 5.1 Limited lookahead approach (LL) for objective LB

procedure LL^{LB}
  $I \leftarrow \lceil n/\kappa \rceil$
  $e^S_{ki} \leftarrow 0 \quad k, i = 1, \ldots, n$
  $z^S_{ij} \leftarrow 0 \quad i, j = 0, \ldots, n, i < j$
  $w^S_{pi} \leftarrow 0 \quad p = 1, \ldots, P, i = 1, \ldots, n$
  $s^S_{pi} \leftarrow 0 \quad p = 1, \ldots, P, i = 1, \ldots, n$
  for $\jmath = 1, \ldots, I$ **do**
    $\underline{n} \leftarrow \kappa(\jmath - 1) + 1$; $\bar{n} \leftarrow \min\{\kappa(\jmath - 1) + \nu, n\}$; $\bar{k} \leftarrow \jmath \cdot \kappa$
    $DefineLB(\jmath)$
    Solve $LB(\jmath)$
    $e^S_{ki} \leftarrow e_{ki} \quad k = \underline{n}, \ldots, \bar{k}, i = 1, \ldots, \bar{n}$
    $z^S_{ij} \leftarrow z_{ij} \quad i, j = 0, \ldots, \bar{n}, i \neq j, \sum_{k=\underline{n}}^{\bar{k}} e^S_{kj} = 1$
    $w^S_{pi} \leftarrow w_{pi}, s^S_{pi} \leftarrow s_{pi} \quad p = 1, \ldots, P, i = 1, \ldots, \bar{n}, \sum_{k=\underline{n}}^{\bar{k}} e^S_{ki} = 1$
  end for
end procedure

Algorithm 5.1 illustrates the procedure for the load balancing objective as an example. At first, we initialize I. Next, solution arrays e^S_{ki}, z^S_{ij}, w^S_{pi} and s^S_{pi} are defined to store solutions of the decision variables of each iteration. The subsequent for-loop describes the iterations of the algorithm. We first initialize the horizon of workpieces to consider in the optimization by defining $\underline{n}$ as a lower and $\bar{n}$ as the upper limit of the respective iteration $\jmath$. $\bar{k}$ serves as the indicator for the last position in the assembly sequence, whose decision variables are to be fixed at the end of the iteration.

Next, procedure $DefineLB(\jmath)$ defines the actual mathematical model (see Algorithm 5.2). The model consists of the basic formulation as defined in previous sections and is altered by replacing the upper limit n by $\bar{n}$ and decision variables e_{ij}, z_{ij}, w_{pi}, s_{pi} by the summation with their stored array counterpart. E.g., in (5.3), z_{ij} is replaced by $(z_{ij} + z^S_{ij})$ resulting in:

$$\sum_{i=0}^{j-1} z_{ij} + z^S_{ij} = 1 \quad j = 1, \ldots, \bar{n}.$$

By doing so, we ensure that the solutions of the previous iterations are fixed in the current iteration. In the example above, if for a given i and j, $z^S_{ij} = 1$, obviously all

Algorithm 5.2 Definition of the mathematical program for $LB(\jmath)$

procedure $DefineLB(\jmath)$

- Let $LB(\jmath)$ be the mathematical program defined by (5.1)–(5.12), (5.20)–(5.23).
- Substitute n with $\bar{n}$ in the variable and constraint domains and inequalities of $LB(\jmath)$.
- In $LB(\jmath)$, substitute e_{ki}, z_{ij}, w_{pi} and s_{pi} by the sum over decision variable and respective saved solution, i.e., with $\left(e_{ki} + e^S_{ki}\right)$, $\left(z_{ij} + z^S_{ij}\right)$, $\left(w_{pi} + w^S_{pi}\right)$ and $\left(s_{pi} + s^S_{pi}\right)$, respectively.

end procedure

decision variables $z_{ik} = 0 \quad k = 1, \ldots, \bar{n}$.

After defining $LB(\jmath)$ in the for loop of Algorithm 5.1, we solve the mathematical model and store its solution in the respective solution arrays, but only up to the last position $\bar{k}$ of the assembly sequence that is fixed. Note that for u_{ij} the decision variables are not saved. If a product is placed on a mix bank in iteration $\jmath$, where workpieces of previous iterations are already fixed, the respective u_{ij} variables are altered for all workpieces on said mix bank, as they represent the number of workpieces to follow. Thus, all u_{ij} are potentially subject to change in each iteration.

Chapter 6.

Conclusion

6.1. Summary and Insights

This thesis provides theoretical advances and practical applications for dealing with uncertainty and fixed costs in inventory and production systems.

First, we proved the optimal inventory control policy when demands are uncertain and fixed costs vary cyclically over time. The resulting policy is complex in the regular order period, where fixed costs are low, but remains a simple period-dependent (s, S) policy in all other periods. We describe and prove this optimal policy based on the notion of K-convexity and the optimal ordering behavior in the presence of non-K-convex cost functions. Although fixed order costs change cyclically, the cost functions of the intraperiods are L-convex and thus lead to optimal (s, S) policies. The regular order period does not fulfill K-convexity, which leads to the more complex order structure with multiple order areas. We show numerically that simpler policies, for instance (s_j, S_j), (s, S, r, R), $(Opt|m)$ and (s, S) policies, provide varying degrees of approximation of the optimal policy. In that regard, the (s, S, r, R) policy leads to average cost deviations < 1 % and the (s_j, S_j) policy to almost optimal solutions (that is: negligible cost differences).

Managers should take away that the optimal policy is not of the simple (s, S) type and that one will make severe errors by applying the policy with the same parameters for all periods. The optimal policy in the regular order period is even more complex than a period-dependent (s, S) policy, but it can be described by a clear structure with multiple order areas. The order areas arise as a forward-buying effect to reduce the probability of ordering in the high-cost periods gains importance. The good news is that, by adding two additional parameters (r and R) to the classic (s, S) policy, the resulting inventory policy is still fairly easy to use yet provides near-optimal solutions in most cases. While this is true for most settings, companies that face little demand variation and small holding costs should be careful, as in this case, a more complex policy might be in order, namely a period-dependent (s, S) policy.

Next, in a practical problem setting at a retailer, we established a model for setting and allocating delivery rhythms and order-up-to levels for products and stores. The complexity of the data-driven model was tackled by the use of hierarchical decomposition approaches in the product or time dimension. Furthermore, a genetic algorithm approach that self-adapts its parameters depending on an instance's characteristics was introduced. Lastly, we established a savings algorithm, which starts with a lot-for-lot solution and, where it is cost-beneficial and feasible, improves by merging orders. We compared the different approaches to find that the genetic algorithm, with the savings algorithm as a starting solution, finds the best results on average. Average cost benefits of over 3 % in a controlled testbed and over 20 % in the case study of a European retailer were achieved.

From a managerial point of view, the results furthermore indicated that the genetic algorithm model is most beneficial when the data-driven model is supplied with a large number of samples and when the time between orders is high. The approach also showed the possibility of including practical constraints such as an upper bound on the workload both in the warehouse and the store, which could easily be extended to a variety of other practice-related constraints without generally altering the model or solution approach.

In a second practical problem from retail, a model to set the parameters for an (s, c, S, nq) inventory control policy with case packs was provided. The data-driven approach incorporates the backroom effect, i.e., the double handling of goods that do not fit on the shelf during initial replenishment. A comparison of the approach with benchmarks revealed cost savings of over 0.9 % when the backroom effect is included, over 17 % when an (s, c, S) policy rather than an (s, S) policy is used and over 53 % when a stochastic data-driven approach rather than one that only optimizes on expected demand values is used.

Hence, managers should take the demand uncertainty as well as the backroom effect into account as they are key cost drivers. If a significant major fixed cost exists, inventory control should be changed to an (s, c, S, nq) policy, which still allows inventory control on an item level, but allows the realization of economies of scale across multiple products.

The third practical problem considered in the thesis stems from an automotive supplier of modules. A model has been introduced to consider resequencing of the assembly sequence at the supplier while ensuring restoration of the OEM sequence through a first-in-first-out approach where mix banks are used. The model formulation has been adapted for the use in three different objectives, namely the load balancing, material leveling and color batching. To make the approach applicable in practice, a limited lookahead heuristic has been developed. The numerical results showed that, on average, > 50 % savings against the production of the OEM sequence as-is can be realized. Furthermore, the heuristic provided good solutions in a short time and has hence proven its applicability to practical-sized problems.

Managers are advised to carefully analyze the proposed model and the parameters for the heuristic by back-testing its capabilities on historical sequences. The largest increase in savings is obtained when the second mix bank is implemented. In general, benefits are high when sequences are neither small nor large, the number of stations/materials/colors is small, more complex products are demanded and the average utilization of stations is not too high.

We summarize our insights to provide answers to our research questions of Chapter 1.

- *How can operations research methods be used to derive advanced decision support in digitized supply chains?*
  In our practical applications, we use data-driven approaches that explicitly utilize given company sales data as-is. Based on our optimization models, we then transform this data into decision support by providing concrete parameter values for the inventory control policies. We use the methodological advantages of MILP models, which are core to many operations research approaches and can be solved by commercial standard solvers. By using the company data, transforming it into decision support and using operations research approaches, we manage to derive large cost savings in our practical applications, which shows the clear benefit of prescriptive analytics in practice.

- *Does the consideration of uncertainty in the decision-making have any impact on the operational performance?*
  We explicitly investigate the effect of including uncertainty in our data-driven models for two practical retail applications. We find that, in both cases, cost benefits are huge when we consider the uncertainty. Moreover, the larger the amount of input data is, the better our approaches. Hence, the current drive to collect more and more data in the industry can effectively be used for reducing

costs considerably.

- *How can innovative operations concepts tackle the requirements for cost-efficiency and practical complexity?*
  In our practical applications, we investigated two advanced delivery concepts in retail and a new organizational approach in production operations for an automotive supplier. We showed that, although the concepts combined with practical constraints might pose difficulties when it comes to their optimal application without proper decision support, our models realize significant savings over those offered by other concepts. For example, the new proposed (s, c, S, nq) policy uses an additional parameter which needs to be optimized, yet it realizes far higher cost benefits than would be possible with the traditional (s, S, nq) policy. Despite the practical requirement of an easy restoration process, our resequencing approach for the automotive supplier leads to vast savings under very different goals. We hence showed that innovative concepts, accompanied by the appropriate decision support tools, can be utilized to realize significant savings in cost, personnel, load variation and other dimensions.

- *What is the implication of various forms of fixed order costs (time-dependent, multi-item or sequence-dependent) for decision-making?*
  Our investigations have shown several implications of fixed order costs. If the fixed order costs are time-dependent, optimal inventory control will likely be of a complex structure. However, simple nearly-optimal inventory policies can approximate the optimal decision-making when the parameters are properly set. The most important factor is an adaptive policy that varies its parameters over time with the variance in the fixed order costs. When we look at a multi-item setting, coordinated inventory control is key when it comes to saving fixed order costs. If no coordination takes place, costs will likely rise severely, even if optimal parameters for the inventory policy are used. In the automotive supplier problem, we investigated sequence-dependent fixed costs in a production setting. Their existence drives the need for flexible production systems immensely, as possible savings are huge. On the other side, the requirements for flexible changes in the production sequence are also significant, even more so if the quality of the customer sequence is poor.

6.2. Further Research

The theoretical and practical problem settings observed in this thesis leave room for future research.

When providing the optimality proof in Chapter 2, we assumed that fixed costs drop once per order cycle. One could investigate a setting where fixed costs vary in all periods of the order cycle to further generalize the model. The other introduced notions mentioned in the chapter all assume non-increasing fixed costs. Consequently, their optimal behavior might be influenced in the same way if those fixed costs, as in our setting, vary over time.

The investigated retail problems all use demand data to optimize inventory control policies. A next extension would be to include causal data to transform the methods into Big Data approaches and further enhance the demand modeling. In that regard, also machine learning approaches, such as regularization, might help to reduce the problem of over-fitting and hence increase the out-of-sample performance.

While the resequencing approach is beneficial in the setting of an automotive supplier, its adaptable formulation could also be transferred to rather different problem settings as, e.g., the roll cage sequencing in retail. In this example, roll cages for the different stores need to be loaded into trucks in sequence, depending on the delivery route. Resequencing might be applied in this setting to alter the sequence in which the picking tours need to occur while allowing the restoration of the delivery sequence at the end. In this case, there would be no need to implement mix banks, as a re-purposing of the outbound area should yield the same result.

The investigation of the practical problem settings was based on rather small numerical test instances. To get closer to practice, larger and more complex data sets should be analyzed. While some of the MILP approaches showed that larger instances will not be solvable, this leaves room for other technical-driven solutions, which should aim at faster computation. Exact approaches such as Benders' decomposition or column generation, as well as mat- or meta-heuristics might be quite applicable to our practical problem settings.

In this thesis, we applied the data-driven approaches on inventory problems in retail. However, the general idea of using historical data in MILP models to generate decision support through optimization has a wide range of applicability. Other industries that experience stochastic demand are also candidates for the approach, but so are other stochastic influences, such as stochastic lead times, yields or prices.

In all of our approaches, we compared several benchmark models and their parameter settings to exact approaches. The numerical studies revealed that for different instances also different approaches perform best. While we identified approaches that perform best across all instances, a further step would be a classifier who selects the most appropriate model for each instance individually, based on the instance's characteristics. Using such a classifier will improve the performance overall and should provide more robustness, i.e. performance outliers should occur less likely.

References

Abdelmaguid, T. F., & Dessouky, M. M. (2006). A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Economics*, *44*(21), 4445–4464.

Adan, I., Van Eenige, M., & Resing, J. (1995). Fitting Discrete Distributions on the First Two Moments. *Probability in the Engineering and Informational Sciences*, *9*(4), 623–632.

Agnetis, A., Pacifici, A., Rossi, F., Lucertini, M., Nicoletti, S., Nicolò, F., ... Pesaro, E. (1997). Scheduling of flexible flow lines in an automobile assembly plant. *European Journal of Operational Research*, *97*, 348–362.

Agrawal, N., & Smith, S. A. (1996). Estimating negative binomial demand for retail inventory management with unobservable lost sales. *Naval Research Logistics*, *43*(6), 839–861.

Aksoy, Y., & Erenguc, S. S. (1988). Multi-item inventory models with co-ordinated replenishments: A survey. *International Journal of Operations & Production Management*, *8*(1), 63–73.

Allouche, F. (2017, February). *Global Sector Report - Retail.* Euler Hermes Economic Research.

Amorim, P., Costa, A. M., & Almada-Lobo, B. (2014). Influence of consumer purchasing behaviour on the production planning of perishable food. *OR Spectrum*, *36*, 669–692.

Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal Inventory Policy. *Econometrica*, *19*(3), 250–272.

Atan, Z., & Erkip, N. (2015). Note on "The Backroom Effect in Retail Operations". *Production and Operations Management*, *24*(11), 1833–1834.

Balintfy, J. L. (1964). On a Basic Class of Multi-Item Inventory Problems. *Management Science*, *10*(2), 287–297.

Ban, G.-Y., & Rudin, C. (2014). The Big Data Newsvendor: Practical Insights from Machine Learning. *MIT Sloan School of Management Working Paper Series*, *5032-13*, 1–52.

Berret, M., Mogge, F., Schlick, T., Fellhauer, E., Söndermann, C., & Schmidt, M. (2016). *Global Automotive Supplier Study 2016.* LAZARD, Roland Berger.

Bertazzi, L., Paletta, G., & Speranza, M. G. (2005). Minimizing the Total Cost in an Integrated Vendor-Managed Inventory System. *Journal of Heuristics*, *11*(5-6), 393–419.

Bertsimas, D., & Doan, X. V. (2010). Robust and data-driven approaches to call centers. *European Journal of Operational Research*, *207*(2), 1072–1085.

Bertsimas, D., & Thiele, A. (2006). A Robust Optimization Approach to Inventory Theory. *Operations Research*, *54*(1), 150–168.

Bertsimas, D., & Thiele, A. (2014). Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty. *INFORMS Tutorials in Operations Research*, 95–122.

Beutel, A.-L., & Minner, S. (2012). Safety stock planning under causal demand forecasting. *International Journal of Production Economics*, *140*(2), 637–645.

Bijvank, M., & Vis, I. (2011). Lost-sales inventory theory: A review. *European Journal of Operational Research*, *215*(1), 1–13.

Boctor, F. F., Laporte, G., & Renaud, J. (2004). Models and algorithms for the dynamic-demand joint replenishment problem. *International Journal of Production Research*, *42*(13), 2667–2678.

Bookbinder, J. H., & Tan, J.-Y. (1988). Strategies for the Probabilistic Lot-Sizing Problem with Service-Level Constraints. *Management Science*, *34*(9), 1096–1108.

Boysen, N., Fliedner, M., & Scholl, A. (2009a). Production planning of mixed-model assembly lines: overview and extensions. *Production Planning & Control*, *20*(5), 455–471.

Boysen, N., Fliedner, M., & Scholl, A. (2009b). Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research*, *192*, 349–373.

Boysen, N., Fliedner, M., & Scholl, A. (2010). Level scheduling under limited resequencing flexibility. *Flexible Services and Manufacturing Journal*, *22*, 236–257.

Boysen, N., Golle, U., & Rothlauf, F. (2011). The Car Resequencing Problem with Pull-Off Tables. *Business Research*, *4*(2), 276–292.

Boysen, N., Scholl, A., & Wopperer, N. (2012). Resequencing of mixed-model assembly lines: Survey and research agenda. *European Journal of Operational Research*, *216*, 594–604.

Boysen, N., & Zenker, M. (2013). A decomposition approach for the car resequencing problem with selectivity banks. *Computers & Operations Research*, *40*, 98–108.

Boywitz, D., Boysen, N., & Briskorn, D. (2016). Resequencing with Parallel Queues to Minimize the Maximum Number of Items in the Overflow Area. *Naval Research Logistics*, *63*(5), 401–415.

Braglia, M., Castellano, D., & Song, D. (2017). Distribution-free approach for stochastic Joint-Replenishment Problem with backorders-lost sales mixtures, and controllable major ordering cost and lead times. *Computers & Operations Research*, *79*, 161–173.

Brauchle, A., Kostron, A., & Schlesner, W. (2015). *Digitization Strategy for Automotive Suppliers: How to systematically utilize chances and avoid risks.* Horváth & Partners.

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (1999). Using Association Rules for Product Assortment Decisions: A Case Study. In *Proceedings of the fifth international conference on knowledge discovery and data mining* (pp. 254–260). San Diego (USA).

Broekmeulen, R. A. C. M., Sternbeck, M. G., Van Donselaar, K. H., & Kuhn, H. (2017). Decision support for selecting the optimal product unpacking location in a retail supply chain. *European Journal of Operational Research*, *259*(1), 84–99.

Broekmeulen, R. A. C. M., & Van Donselaar, K. H. (2004). Excess shelf space in retail stores: An analytical model and empirical assessment. *BETA publicatie: working papers, Technische Universiteit Eindhoven, Eindhoven*, *109*, 1–26. Retrieved from `https://pure.tue.nl/ws/files/2406788/587322.pdf`

Bughin, J., LaBerge, L., & Mellbye, A. (2017). The case for digital reinvention. *McKinsey Quarterly*, *February*, 1–19.

Campbell, A. M., & Savelsbergh, M. W. P. (2004). A Decomposition Approach for the Inventory-Routing Problem. *Transportation Science*, *38*(4), 488–502.

Cárdenas-Barrón, L. E., González-Velarde, J. L., & Treviño-Garza, G. (2015). A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem. *Computers & Operation Research*, *64*, 225–232.

Chen, V. C. P., Tsui, K.-L., Barton, R. R., & Meckesheimer, M. (2006). A review on design, modeling and applications of computer experiments. *IIE Transactions*, *38*(4), 273–291.

Chen, X., & Simchi-Levi, D. (2004). Coordinating Inventory Control and Pricing Strategies with Random Demand and Fixed Ordering Cost: The Finite Horizon Case. *Operations Research*, *52*(6), 887–896.

Chiang, C. (2003). Optimal replenishment for a periodic review inventory system with two supply modes. *European Journal of Operational Research*, *149*(1), 229–244.

Chiang, C. (2006). Optimal ordering policies for periodic-review systems with replenishment cycles. *European Journal of Operational Research*, *170*(1), 44–56.

Chica, M., Bautista, J., Cordón, Ó., & Damas, S. (2016). A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega*, *58*, 55–68.

Chick, S. E., Huchzermeier, A., & Netessine, S. (2014). Europe's solution factories. *Harvard Business Review*, *92*(4), 111–115.

Choi, W., & Shin, H. (1997). A Real-Time Sequence Control System for the Level Production of the Automobile Assembly Line. *Computers & Industrial Engineering*, *33*(3-4), 769–772.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, *39*(11), 2537–2548.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2013). Thirty Years of Inventory Routing. *Transportation Science*, *48*(1), 1–19.

Cordeau, J.-F., Laganà, D., Musmanno, R., & Vocaturo, F. (2014). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, *55*, 153–166.

Corsten, D., & Gruen, T. (2003). Desperately seeking shelf availability: an examination of the extent, the causes, and the efforts to address retail out-of-stocks. *International Journal of Retail & Distribution Management*, *31*(12), 605–617.

Curşeu, A., Van Woensel, T., Fransoo, J. C., Van Donselaar, K. H., & Broekmeulen, R. A. C. M. (2008). Modelling handling operations in grocery retail stores: an empirical analysis. *Journal of the Operational Research Society*, *60*(2), 200–214.

DeHoratius, N., & Ton, Z. (2009). The Role of Execution in Managing Product Availability. In N. M. Agrawal & S. A. Smith (Eds.), *Retail supply chain management* (pp. 53–77). New York: Springer.

Digital Supply Chains: Increasingly Critical for Competitive Edge. (2015). AT Kearney and WHU. Retrieved from `https://www.atkearney.com/documents/10192/6500433/Digital+Supply+Chains.pdf`

Ding, F.-Y., & Sun, H. (2004). Sequence alteration and restoration related to sequenced parts delivery on an automobile mixed-model assembly line with multiple departments. *International Journal of Production Research*, *42*(8), 1525–1543.

Doran, D. (2001). Synchronous supply: an automotive case study. *European Business Review*, *13*(2), 114–120.

Doran, D. (2002). Manufacturing for synchronous supply: a case study of Ikeda Hoover Ltd. *Integrated Manufacturing Systems*, *13*(1), 18–24.

Dörmer, J., Günther, H.-O., & Gujjula, R. (2015). Master production scheduling and sequencing at mixed-model assembly lines in the automotive industry. *Flexible Services and Manufacturing Journal*, *27*, 1–29.

Drummond, L., Ochi, L. S., & Vianna, D. S. (2001). An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, *17*, 379–386.

Dunke, F., & Nickel, S. (2016). A general modeling approach to online optimization with lookahead. *Omega*, *63*, 134–153.

East, R., Lomax, W., Willson, G., & Harris, P. (1994). Decision Making and Habit in Shopping Times. *European Journal of Marketing*, *28*(4), 56–71.

Ehmke, J. F., Campbell, A. M., & Thomas, B. W. (2016). Data-driven approaches for emissions-minimized paths in urban areas. *Computers & Operation Research*, *67*, 34–47.

Ehrenthal, J. C. F., Honhon, D., & Van Woensel, T. (2014). Demand seasonality in retail inventory management. *European Journal of Operational Research*, *238*(2), 527–539.

Epping, T., & Hochstättler, W. (2003). Sorting with Line Storage Systems. In U. Leopold-Wildburger, F. Rendl, & G. Wäscher (Eds.), *Operations research proceedings 2002* (pp. 235–240). Berlin: Springer.

Eroglu, C., Williams, B. D., & Waller, M. A. (2011). Consumer-driven retail operations: The moderating effects of consumer demand an case pack quantity. *International Journal of Physical Distribution & Logistics Management*, *41*(5), 420–434.

Eroglu, C., Williams, B. D., & Waller, M. A. (2013). The Backroom Effect in Retail Operations. *Production and Operations Management*, *22*(4), 915–923.

Federgruen, A., Groenevelt, H., & Tijms, H. C. (1984). Coordinated replenishments in a multi-item inventory system with compound Poisson demands. *Management Science*, *30*(3), 344–357.

Feng, H., Wu, Q., Muthuraman, K., & Deshpande, V. (2015). Replenishment Policies for Multi-Product Stochastic Inventory Systems with Correlated Demand and Joint-Replenishment Costs. *Production and Operations Management*, *24*(4), 647–664.

Feng, Y., & Sun, J. (2001). Computing the Optimal Replenishment Computing Policy for Inventory Systems With Random Discount Opportunities. *Operations Research*, *49*(5), 790–795.

Franz, C., Hällgren, E. C., & Koberstein, A. (2014). Resequencing orders on mixed-model assembly lines: Heuristic approaches to minimise the number of overload situations. *International Journal of Production Research*, *52*(19), 5823–5840.

Fredriksson, P. (2006). Operations and logistics issues in modular assembly processes: cases from the automotive sector. *Journal of Manufacturing Technology Management*, *17*(2), 168–186.

Gallego, G., & Sethi, S. P. (2005). K-Convexity in R^n. *Journal of Optimization Theory and Applications*, *127*(1), 71–88.

Gao, L. L., Altay, N., & Robinson, E. P. (2008). A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand. *Mathematical and Computer Modelling*, *47*(11-12), 1254–1263.

Gaur, V., & Fisher, M. L. (2004). A Periodic Inventory Routing Problem at a Supermarket Chain. *Operations Research*, *52*(6), 813–822.

Gavirneni, S., & Morton, T. E. (1999). Inventory control under speculation: Myopic heuristics and exact procedures. *European Journal of Operational Research*, *117*, 211–221.

Gavish, B., & Graves, S. C. (1978). The travelling salesman problem and related problems. *Massachusetts Institute of Technology, Operations Research Working Paper*, *OR 078-78*.

Gibson, B., Defee, C., & Ishfaq, R. (2016). *Sixth Annual State of Retail Supply Chain Report.* (Auburn University Research Report)

Goyal, S. K., & Satir, A. T. (1989). Joint replenishment inventory control: deterministic and stochastic models. *European Journal of Operational Research*, *38*, 2–13.

Groer, C., Golden, B., & Wasil, E. (2009). The Consistent Vehicle Routing Problem.

Manufacturing & Service Operations Management, *11*(4), 630–643.

Gujjula, R., & Günther, H.-O. (2009). Resequencing Mixed-Model Assembly Lines under Just-In-Sequence constraints. In *International conference on computers and industrial engineering (cie 2009)* (pp. 668–673). IEEE.

Han, Y.-H., & Zhou, C. (2010). Dynamic sequencing of jobs on conveyor systems for minimizing changeovers. *The International Journal of Advanced Manufacturing Technology*, *49*, 1251–1259.

Holzapfel, A., Hübner, A. H., Kuhn, H., & Sternbeck, M. G. (2016). Delivery pattern and transportation planning in grocery retailing. *European Journal of Operational Research*, *252*, 54–68.

Hua, Z., Yang, J., Huang, F., & Xu, X. (2009). A static-dynamic strategy for spare part inventory systems with nonstationary stochastic demand. *Journal of the Operational Research Society*, *60*(9), 1254–1263.

Hübner, A. H., Kuhn, H., & Sternbeck, M. G. (2013). Demand and supply chain planning in grocery retail: an operations planning framework. *International Journal of Retail & Distribution Management*, *41*(7), 512–530.

Hübner, A. H., & Schaal, K. (2017). Effect of replenishment and backroom on retail shelf-space planning. *Business Research*, *10*, 123–156.

Huh, W. T., Levi, R., Rusmevichientong, P., & Orlin, J. B. (2011). Adaptive Data-Driven Inventory Control with Censored Demand Based on Kaplan-Meier Estimator. *Operations Research*, *59*(4), 929–941.

Hurter, A. P., & Kaminsky, F. C. (1968). Inventory Control with a Randomly Available Discount Purchase Price. *Operations Research*, *19*(4), 433–444.

Iglehart, D. L. (1963). Optimality of (s, S) policies in the infinite horizon dynamic inventory problem. *Management Science*, *9*(2), 259–268.

Inman, R. R., & Schmeling, D. M. (2003). Algorithm for agile assembling-to-order in the automotive industry. *International Journal of Production Research*, *41*(16), 3831–3848.

Islamoglu, N. E., Ryu, K., & Moon, I. (2014). Labour productivity in modular assembly: a study of automotive module suppliers. *International Journal of Production Research*, *52*(23), 6954–6970.

Iyer, A. V., & Schrage, L. E. (1992). Analysis of the deterministic (s, S) inventory problem. *Management Science*, *38*(9), 1299–1313.

Jaillet, P., Bard, J. F., Huang, L., & Dror, M. (2002). Delivery Cost Approximations for Inventory Routing Problems in a Rolling Horizon Framework. *Transportation Science*, *36*(3), 292–300.

Jain, A., Groenevelt, H., & Rudi, N. (2011). Periodic Review Inventory Management with Contingent Use of Two Freight Modes with Fixed Costs. *Naval Research Logistics*, *58*(4), 400–409.

Johansen, S. G., & Melchiors, P. (2003). Can-order policy for the periodic-review joint replenishment problem. *Journal of the Operational Research Society*, *54*(3), 283–290.

Johansen, S. G., & Thorstenson, A. (2014). Emergency orders in the periodic-review inventory system with fixed ordering costs and compound Poisson demand. *International Journal of Production Economics*, *157*, 147–157.

Kahn, B. E., & Schmittlein, D. C. (1989). Shopping Trip Behavior: An Empirical Investigation. *Marketing Letters*, *1*(1), 55–69.

Khouja, M., & Goyal, S. K. (2008). A review of the joint replenishment problem literature: 1989-2005. *European Journal of Operational Research*, *186*(1), 1–16.

Kiesmüller, G. P., & Broekmeulen, R. A. C. M. (2010). The benefit of VMI strategies in a stochastic multi-product serial two echelon system. *Computers & Operations Research*, *37*(2), 406–416.

Kirca, Ö. (1995). A primal-dual algorithm for the dynamic lotsizing problem with joint set-up costs. *Naval Research Logistics*, *42*(5), 791–806.

Ko, S.-S., Han, Y.-H., & Young, J. (2016). Paint batching problem on M-to-1 conveyor systems. *Computers & Operation Research*, *74*, 118–126.

Kouki, C., Babai, M. Z., Jemai, Z., & Minner, S. (2016). A coordinated multi-item inventory system for perishables with random lifetime. *International Journal of Production Economics*, *181*, 226–237.

Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2014). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, *63*(1), 60–81.

Kuhn, H., Wensing, T., & Sternbeck, M. G. (2015). Evaluation of case-pack sizes in grocery retailing using a Markov Chain approach. In G. Liberopoulos (Ed.), *10th conference on stochastic models of manufacturing and service operations (smmso 2015)* (pp. 103–110). Volos, Greece: University of Thessaly Press.

Lahmar, M., & Benjaafar, S. (2007). Sequencing with limited flexibility. *IIE Transactions*, *39*(10), 937–955.

Lahmar, M., Ergan, H., & Benjaafar, S. (2003). Resequencing and Feature Assignment on an Automated Assembly Line. *IEEE Transactions on Robotics and Automation*, *19*(1), 89–102.

Law, A. M. (2015). *Simulation Modeling and Analysis* (5th ed.). New York: Mc Graw Hill.

Lim, A., & Xu, Z. (2009). Searching optimal resequencing and feature assignment on an automated assembly line. *The Journal of the Operational Research Society*, *60*(3), 361–371.

Lin, L., Sun, H., & Xu, Y.-Q. (2011). A Research of Resequencing Problem in Automobile Paint Shops Using Selectivity Banks. In *Proceedings of the 2011 ieee 18th international conference on industrial engineering and engineering management* (pp. 658–662). Changchun, China.

Löhndorf, N., & Minner, S. (2013). Simulation optimization for the stochastic economic lot scheduling problem. *IIE Transactions*, *45*(7), 796–810.

Lutfe, M. M., Rajpurohit, K., Rosenberger, J. M., Zaruba, G., & Priest, J. (2015). Optimizing real-time vehicle sequencing of a paint shop. *Omega*, *55*, 61–72.

Martel, A., Diaby, M., & Boctor, F. F. (1995). Multiple items procurement under stochastic nonstationary demands. *European Journal of Operational Research*, *87*(1), 74–92.

Mehrotra, M., Dawande, M., Gavirneni, S., Demirci, M., & Tayur, S. (2011). OR PRACTICE–Production Planning with Patterns: A Problem from Processed Food Manufacturing. *Operations Research*, *59*(2), 267–282.

Meyr, H. (2004). Supply chain planning in the German automotive industry. *OR Spectrum*, *26*(4), 447–470.

Michalewicz, Z., & Schmidt, M. (2007). *Adaptive Business Intelligence*. Berlin, Heidelberg, New York: Springer.

Minner, S. (2009). A comparison of simple heuristics for multi-product dynamic demand lot-sizing with limited warehouse capacity. *International Journal of Production Economics*, *118*, 305–310.

Moin, N. H., & Salhi, S. (2006). Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, *58*(9), 1185–1194.

Moin, N. H., Salhi, S., & Aziz, N. A. B. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics*, *133*(1), 334–343.

Mortenson, M. J., Doherty, N. F., & Robinson, S. (2015). Operational research from Taylorism to Terabytes: A research agenda for the analytics age. *European Journal of Operational Research*, *241*(3), 583–595.

Nagasawa, K., Irohara, T., Matoba, Y., & Liu, S. (2015). Applying Genetic Algorithm for Can-Order Policies in the Joint Replenishment Problem. *Industrial Engineering & Management Systems*, *14*(1), 1–10.

Ohno, K., Tomonori, I., & Toshiaki, Y. (1994). A New Algorithm for a Multi-item Periodic Review Inventory System. *Mathematical Methods of Operations Research*, *39*, 349–364.

Pil, F. K., & Holweg, M. (2004). Linking Product Variety to Order-Fulfillment Strategies. *Interfaces*, *34*(5), 394–403.

Pires, M., Amorim, P., Liz, J., & Pratas, J. (2015). Design of retail backroom storage: A research opportunity? In *Operations research and big data* (pp. 167–174). Springer.

Porteus, E. L. (1971). On the Optimality of Generalized (s,S) Policies. *Management Science*, *17*(7), 411–426.

Porteus, E. L. (2002). *Foundations of stochastic inventory theory* (E. L. Porteus, Ed.). Stanford: Stanford University Press.

Raa, B., & Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, *192*(2), 429–441.

Raman, A., DeHoratius, N., & Ton, Z. (2001). Execution: The missing link in retail operations. *California Management Review*, *43*(3), 136–152.

Reiner, G., Teller, C., & Kotzab, H. (2013). Analyzing the Efficient Execution of In-Store Logistics Processes in Grocery Retailing-The Case of Dairy Products. *Production and Operations Management*, *22*(4), 924–939.

Rigby, D. K. (2017, September 15). The AmazonWhole Foods Deal Means Every Other Retailer's Three-Year Plan Is Obsolete. *Harvard Business Review Digital Articles*.

Sachs, A.-L., & Minner, S. (2014). The data-driven newsvendor with censored demand observations. *International Journal of Production Economics*, *149*, 28–36.

Saliby, E. (1990). Descriptive sampling: a better approach to Monte Carlo simulation.

Journal of the Operational Research Society, *41*(12), 1133–1142.

Santner, T. J., Williams, B. J., & Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. New York: Springer.

Scarf, H. (1959). The Optimality of (S,s) Policies in the Dynamic Inventory Problem. In K. J. Arrow, S. Karlin, & H. Scarf (Eds.), *Mathematical methods in the social sciences* (pp. 196–202). Stanford: Stanford University Press.

Schrauf, S., & Berttram, P. (2016, September 7). Industry 4.0: How digitization makes the supply chain more efficient, agile, and customer-focused. *strategy&*, pp. 1–32. Retrieved from `https://www.strategyand.pwc.com/reports/industry4.0`

Semple, J. (2007). Note: Generalized Notions of Concavity with an Application to Capacity Management. *Operations Research*, *55*(2), 284–291.

Sethi, S. P., & Cheng, F. (1997). Optimality of (s, S) policies in inventory models with Markovian demand. *Operations Research*, *46*(6), 931–939.

Sharma, A. (2017, July 21). How Retail Can Thrive in a World Without Stores. *Harvard Business Review Digital Articles*.

Silver, E. A. (1974). A control system for coordinated inventory replenishment. *International Journal of Production Research*, *12*(6), 647–671.

Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (3rd ed.). New York: Wiley.

Song, J.-S., & Zipkin, P. (1993). Inventory control in a fluctuating demand environment. *Operations Research*, *41*(2), 351–370.

Spieckermann, S., Gutenschwager, K., & Voß, S. (2004). A sequential ordering problem in automotive paint shops. *International Journal of Production Research*, *42*(9), 1865–1878.

Sternbeck, M. G. (2015). A store-oriented approach to determine order packaging quantities in grocery retailing. *Journal of Business Economics*, *85*, 569–596.

Sternbeck, M. G., & Kuhn, H. (2014). An integrative approach to determine store delivery patterns in grocery retailing. *Transportation Research Part E: Logistics and Transportation Review*, *70*, 205–224.

Sun, H., Fan, S., Shao, X., & Zhou, J. (2015). A colour-batching problem using selectivity banks in automobile paint shops. *International Journal of Production Research*, *53*(4), 1258–1276.

Swaminathan, J. M., & Nitsch, T. R. (2007). Managing product variety in automobile assembly: The importance of the sequencing point. *Interfaces*, *37*(4), 324–333.

Taube, F., & Minner, S. (2017a). Data-driven assignment of delivery patterns in retail with handling effort considerations. *Computers & Operations Research*, 1–15. (Forthcoming) doi: https://dx.doi.org/10.1016/j.cor.2017.08.004

Taube, F., & Minner, S. (2017b). *Optimal inventory control with cyclic fixed order costs.* Technical University Munich. (Working Paper)

Taube, F., & Minner, S. (2017c). Resequencing mixed-model assembly lines with restoration to customer orders. *Omega*, 1–13. (Forthcoming) doi: https://dx.doi.org/10.1016/j.omega.2017.11.006

Tempelmeier, H., & Hilger, T. (2015). Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research*, *59*, 119–125.

Tijms, H. C. (2003). *A First Course in Stochastic Models.* Chichester: Wiley.

Ton, Z., & Raman, A. (2010). The Effect of Product Variety and Inventory Levels on Retail Store Sales: A Longitudinal Study. *Production and Operations Management*, *19*(5), 546–560.

Toth, P., & Vigo, D. (2014). *Vehicle Rounting: Problems, Methods, and Applications* (2nd ed.). Bologna, Italy: Society for Industrial and Applied Mathematics and the Mathematical Optimization Society.

Tsai, C.-Y., Tsai, C.-Y., & Huang, P.-W. (2009). An association clustering algorithm for can-order policies in the joint replenishment problem. *International Journal of Production Economics*, *117*, 30–41.

Turgut, Ö., Taube, F., & Minner, S. (2017). *Data-driven retail inventory management with backroom effects.* Technical University Munich. (Working Paper)

Valero-Herrero, M., Garcia-Sabater, J. P., Vidal-Carreras, P. I., & Canos-Daros, L. (2014). Solving the Car Resequencing Problem with mix banks. *Dirección y Organización*, *54*, 36–44.

Van Donselaar, K. H., & Broekmeulen, R. A. C. M. (2013). Determination of safety stocks in a lost sales inventory system with periodic review, positive lead-time, lot-sizing and a target fill rate. *International Journal of Production Economics*, *143*(2), 440–448.

Van Donselaar, K. H., Gaur, V., Van Woensel, T., Broekmeulen, R. A. C. M., & Fransoo, J. C. (2010). Ordering Behavior in Retail Stores and Implications for Automated

Replenishment. *Management Science*, *56*(5), 766–784.

Van Donselaar, K. H., Van Woensel, T., Broekmeulen, R. A. C. M., & Fransoo, J. C. (2006). Inventory control of perishables in supermarkets. *International Journal of Production Economics*, *104*(2), 462–472.

Van Woensel, T., Erkip, N., Curşeu, A., & Fransoo, J. C. (2013). Lost Sales Inventory Models with Batch Ordering and Handling Costs. *BETA publicatie: working papers, Technische Universiteit Eindhoven, Eindhoven*, *421*, 1–34. Retrieved from http://cms.ieis.tue.nl/Beta/Files/WorkingPapers/wp_421.pdf

Van Zelst, S., Van Donselaar, K. H., Van Woensel, T., Broekmeulen, R. A. C. M., & Fransoo, J. C. (2009). Logistics drivers for shelf stacking in grocery retail stores: Potential for efficiency improvement. *International Journal of Production Economics*, *121*(2), 620–632.

Veinott, A. F. (1966). On the Opimality of (s,S) Inventory Policies: New Conditions and a New Proof. *SIAM Journal on Applied Mathematics*, *14*(5), 1067–1083.

Veinott, A. F., & Wagner, H. M. (1965). Computing optimal (s, S) inventory policies. *Management Science*, *11*(5), 525–552.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, *60*(3), 611–624.

Waller, M. A., Heintz, A., & Brent, T. (2008). Case pack quantity's effect on retail market share: An examination of the backroom logistics effect and the store-level fill rate effect. *International Journal of Physical Distribution & Logistics Management*, *38*(6), 436–451.

White, K. P. (1997). An Effective Truncation Heuristic for Bias Reduction in Simulation Output. *Simulation*, *69*(6), 323–334.

Ye, Q., & Duenyas, I. (2007). Optimal Capacity Investment Decisions with Two-Sided Fixed-Capacity Adjustment Costs. *Operations Research*, *55*(2), 272–283.

Zipkin, P. (1989). Critical Number Policies for Inventory Models with Periodic Data. *Management Science*, *35*(1), 71–80.