MASTER

Transient optimization of capacitated supply chain operations planning with lost sales including applications to an EyeOn case study

van Eekelen, W.J.E.C.

Award date:
2019

Department of Industrial Engineering and Innovation Sciences
Department of Mathematics and Computer Science

Transient Optimization of Capacitated Supply Chain Operations Planning with Lost Sales

including applications to an EyeOn case study

W.J.E.C. van Eekelen

BSc. in Industrial Engineering and Management Sciences

In partial fulfillment of the requirements for the degree of
MSc. in Operations Management and Logistics
and the degree of
MSc. in Industrial and Applied Mathematics

Supervisors:

prof.dr. A.G. de Kok	-	OPAC group
prof.dr.ir. S.C. Borst	-	SOR group
dr.ir. M.M. Jansen	-	EyeOn

Eindhoven, July 30, 2019

Eindhoven University of Technology
School of Industrial Engineering and Management Sciences
School of Mathematics
Series Master Theses Operations Research

Keywords: Stochastic MLCLSP, SCOP, Robust optimization, Stochastic programming, Production-inventory systems, Capacitated, Lost Sales

Abstract

In this study, we resolve a capacitated production-inventory problem with uncertain demand and lost sales. To this purpose, we use several different optimization approaches. These methods include robust optimization, sample average approximation, and distributionally robust optimization. Our contribution to literature is twofold. First of all, we emphasize the relevance of our research for supply chain planning problems in industry. In essence, the problem defined is the stochastic Multi-Level Capacitated Lot-Sizing Problem. Note that no variant of this problem has ever been subject to a study comparing the methodologies used in this thesis. Secondly, all production-inventory models discussed in previous research assume backordering. To our knowledge, we are the first to consider lost-sales inventory models for robust optimization approaches. The aforementioned models were derived, analyzed, and compared in this thesis. For this comparison an extensive Monte Carlo simulation was performed. In general, the simulation shows the relevance of modeling uncertainty of demand explicitly when solving a stochastic Multi-Level Capacitated Lot-Sizing Problem. On average, the relative improvement over the deterministic model hit a staggering 30%. Based on the results of the study, it can be concluded that the underlying planning behaviour of the different methods conforms to the general phenomena that are observed for these specific optimization paradigms. Finally, we provide insights on the scalability of our models and present an application of our methods in a practical case study.

Executive Summary

Introduction

Tactical capacitated production-inventory planning under uncertainty is one of the most important problems in Operations Research & Management Sciences. Production scheduling and inventory decision making under uncertainty is an important element of supply chain management due to the induced supply chain risks such as, i.a., lost sales and abundant inventory. Considering demand uncertainty in the planning process is essential in order to mitigate the aforementioned risks. The greater part of literature considers the deterministic counterpart of these type of problems. However, our world is inherently stochastic and thus requires planning approaches that acknowledge this uncertainty.
Many production processes are subject to capacity constraints. Therefore, order release decisions should be taken to satisfy future and current demand with timely produced stock against minimum costs. These decisions consider both the coordination of material and the allocation of available resource capacity.
We describe our problem with the following outline:

- We consider the short-term optimization of the supply chain under consideration. Hence, we study the transient behaviour of the production-inventory system. We consider monthly time buckets and optimize a planning horizon of 6-12 months in a rolling-horizon scheme fashion, which is common in practice. Initial inventory levels have a significant impact on the transient behaviour of these systems.
- Demand which cannot be fulfilled from on-hand inventory is lost. Thus, our system considers lost sales. Our models focus on the minimization of lost sales and inventory holding costs. Hence, they implicitly maximize the expected profit.
- Demand is uncertain. We assume that forecasts exist which provide the actual values of the expected demand and the coefficient of variation of the end items. Furthermore, we have full information on the underlying probability distribution of the demand.
- The underlying supply chain of the production-inventory system can be of a general structure.

In our research, we will deal with this supply chain planning problem on the tactical planning level. This problem is approached from different angles. Several sophisticated solution approaches are derived, analyzed and compared. These methods include robust optimization, sample average approximation, and distributionally robust optimization.
Our contribution is twofold. First of all, we emphasize the relevance of our research for supply chain planning problems in industry. In essence, the problem defined is the stochastic Multi-Level Capacitated Lot-Sizing Problem. Note that no variant of the this problem has ever been subject to a study comparing the methodologies used in this thesis. Secondly, all production-inventory models discussed in previous research assume backordering. To our knowledge, we are the first to consider lost-sales inventory models for robust optimization approaches.

Models

We used the following types of mathematical programming models to solve the aforementioned problem:

1. Deterministic LP-based rolling-horizon planning. This is our benchmark optimization model. Many planners in practice claim that solving these LP problems generates the true optimal result. However, our world is inherently uncertain, and hence it is necessary to model certain input parameters as stochastic processes. This optimization model resolves the situation were the demand is assumed to be known in advance. In our variant of the model, the demand parameter is set to the expectation of demand.

2. Safety stock heuristic. This is an enhanced reformulation of the benchmark LP model. This model incorporates the uncertainty in demand by targeting a required safety stock level. We use newsvendor type of equations to compute these safety stocks. However, these do not hold in the lost-sales, capacitated planning problem that we are considering. Nevertheless, it provides us with a heuristic that incorporates the uncertainty in demand.

3. Robust optimization (RO). This method approaches modeling the uncertainty of demand in a different way. The uncertainty of the demand is captured by an uncertainty set, and the worst-case performance of the system is minimized over this entire set. Hence, the uncertainty is captured in a geometric fashion. We define the worst-case performance as the maximum cost incurred during our Monte Carlo simulation.

4. Sample average approximation (SAA). This is an approximate method for solving the exact multi-stage stochastic programming model that is related to the problem that we defined above. Note that this stochastic programming model represents the true formalization of our problem.

5. Distributionally robust optimization (DRO). This method optimizes the problem with respect to the worst-case expected performance. Hence, this approach positions itself in an area between the geometric approach of RO and the stochastic solution method SAA.

Results

Table 1: VMU, VSP, and VRO for the different methods (all cases included)

	VMU	VSP	VRO	p-value DRO difference
LP	0.00%	41.57%	29.67%	< 0.001***
Safety stock	28.97%	0.46%	4.97%	< 0.001***
RO	29.40%	-0.14%	0.00%	< 0.001***
SAA	29.30%	0.00%	4.16%	< 0.001***
DRO	30.23%	-1.33%	1.40%	

*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Table 1 shows us some of the results that we found. VMU is the relative improvement of our methods over the deterministic LP, VSP describes the improvement of SAA over the other methods, and VRO captures the value of using RO in the worst-case situations. It turned out that the SAA approach does not perform significantly better than the other models that incorporate uncertainty. Overall, the SAA method outperformed the RO and safety stock approaches, but the former was beaten by the DRO model. Theoretically, SAA should have outperformed all the other methods on average, as the stochastic programming paradigm is grounded upon minimization of expectation values. However, we were dealing with heuristic/approximate procedures here. First of all, there existed a discrepancy between the assumed

distribution and theoretical distribution. Hence, the scenario generation procedure of SAA was biased. DRO is more robust to distribution fitting errors, as it only considers four specific parameters of the underlying distribution: the lower/upper bound of the support,the mean of the r.v., and its mean absolute deviation from the mean. Moreover, SAA is an approximate solution procedure for the original stochastic program. These could be reasons, i.a., that DRO was able to outperform SAA. The results of the RO model were in accordance with the RO paradigm. This model indeed minimized the maximum cost for almost all cases. In general, the VRO is clearly present. However, RO is subject to a trade off between performance and conservatism. Therefore, it is also possible to generate solution with a good average cost.
Moreover, some additional experiments were performed to assess the scalability of our methods in the amount of items. It turned out that the SAA approach has a hard time scaling due to the amount of binary variables increasing substantially when introducing new items. Distributionally robust optimization, however, still runs in acceptable CPU times whilst returning the best solutions.

Conclusion

First of all, we emphasize the relevance of our research for supply chain planning problems in industry. In essence, the problem defined is the stochastic Multi-Level Capacitated Lot-Sizing Problem. Our model incorporates a significant set of issues that occur in practice. Furthermore, we are the first to construct models that are able to solve lost-sales inventory problems by using robust optimization techniques.
In general, our research shows the value of modeling uncertainty of demand explicitly when solving a stochastic Multi-Level Capacitated Lot-Sizing Problem. On average, the relative improvement over the deterministic case hit a staggering 30%. To assess the advantages/disadvantages of the other optimization methods, we constructed an extensive Monte Carlo simulation study. Based on the results of the study, it can be concluded that the underlying planning behaviour of the different methods conforms to the general phenomena that are observed for these specific optimization paradigms. Robust optimization tends to expand the inventory buffer to mitigate the risk of high penalty costs. Sample average approximation has the propensity to focus on the expected costs, and distributionally robust optimization possesses the aptitude to protect against the worst-case expected behaviour. Hence, in general distributionally robust optimization maintains higher inventory levels than SAA, and therefore the former provides higher customer service levels.
The experiments also provided us with valuable computational insights. First of all, it turns out that the performance of the SAA approach deteriorates considerably when the number of items increases. This method is unable to provide provable optimal results to the MIP model. Moreover, the incumbent solution that is returned is inferior to the solutions that RO and DRO provide. Distributionally robust optimization seems to outperform all the other methods. However, note that we only scaled the problem in the amount of items. Increasing the number of planning periods would probably affect the cost/computation time performance of the latter method considerably as well.

Preface

This thesis is the the final product that I will deliver as a student of the TU/e and the first stepping stone to my career as a PhD researcher. It contains nine months of dedicated research, where I tried to capture both theoretical and practical insights. The gray area overlapping both combinatorial optimization and stochastic operations research is my prospective research topic, and it is already ubiquitous in this thesis.
It has been hard work during the last years. So I would like to thank my parents, who gave me the possibility to pursue my studies and supported me throughout. Furthermore, I would like to thank my friends for the fun and laughter during the past six years.
There are also several people involved with this thesis that I would like to thank. First of all, I want to thank my mentor Ton de Kok for providing invaluable feedback during our meetings. When I got stuck deriving one of the most important models in this thesis, you pointed me to the right expression that I could use to resolve the issues I had encountered. Furthermore, I would like to thank my second mentor/supervisor, Sem Borst, for his thorough analysis of my thesis and the related critical notes. Finally, a thank you is in place for my company supervisor, Michiel Jansen, who also provided me with plenty of valuable insights and helped preserve the link of my research to industry.

Wouter van Eekelen

> *"The mind, once stretched by a new idea, never returns to its original dimensions."*
>
> \- Ralph Waldo Emerson

Contents

List of Figures

List of Tables

Abbreviations

APS Advanced planning software

ARO Adjustable robust optimization

B&B Branch-and-Bound

BOM Bill-of-Material

CLSP Capacitated Lot-Sizing Problem

CODP Customer Order Decoupling Point

DRO Distributionally robust optimization

ELSP Economic Lot-Sizing Problem

EOQ Economic Order Quantity

GLSP General Lot-Sizing and Scheduling Problem

I&L Inventory and lot-size

LP Linear programming

MAD Mean absolute deviation from the mean

MIP Mixed integer programming

MLCLSP Multi-Level Capacitated Lot-Sizing Problem

MLCLSPL Multi-Level Capacitated Lot-Sizing Problem with Linked lot sizes

MOQ Minimum order quantity

MP Mathematical programming

MRP Manufacturing Resource Planning

PU Production unit

RC Robust counterpart

RO Robust optimization

S-MLCLSP Stochastic Multi-Level Capacitated Lot-Sizing Problem

SAA Sample average approximation

SCOP Supply Chain Operations Planning

SKU Stock-keeping unit

SPLP Simple Plant Location Problem

SRP Shortest Route Problem

VMU Value of Modeling Uncertainty

VRO Value of Robust Optimization

VSP Value of Stochastic Programming

Chapter 1

Introduction

Tactical capacitated production-inventory planning under uncertainty is one of the most important problems in Operations Research & Management Sciences. Production scheduling and inventory decision making under uncertainty is an important element of supply chain management due to the induced supply chain risks such as, i.a., lost sales and abundant inventory. Considering demand uncertainty in the planning process is essential in order to mitigate the aforementioned risks (Guillaume, Thierry, & Zieliński, 2017). The greater part of literature considers the deterministic counterpart of these type of problems. However, our world is inherently stochastic and thus requires planning approaches that acknowledge this uncertainty. In our research, we will deal with a supply chain planning problem on the tactical planning level. This study encompasses a general multi-level, multi-item, multi-resource model with random demand and lost sales. This problem is approached from different angles. Several sophisticated solution approaches are derived, analyzed and compared. These methods include, i.a., robust optimization (RO), sample average approximation (SAA), and distributionally robust optimization (DRO). The measures of interest are cost performance, customer service levels and computation times. We try to optimize the short-term performance of the system, i.e., we study its transient behaviour. As exact optimization of the underlying problem is restricted by the curse of dimensionality, we classify our methods as heuristic/approximate procedures founded on well-known mathematical optimization techniques.

1.1 Outline

The general setup of this thesis is as follows. First of all, we provide a comprehensive description of the problem context in Section 1.2, which includes a discussion on the practical relevance of our problem. Section 1.3 provides the theoretical framework used and a general setup of the research design. Furthermore, this section states the formal model that defines our problem, including the scope and assumptions, a small outline of the different optimization techniques, and a discussion on the relevant contributions to literature. In Chapter 2 we provide a comprehensive literature review on the topics relevant to our research. Chapter 3 describes the different optimization models and the derivations that were necessary to construct them. We describe the design of experiments and the Monte Carlo simulation study in Chapter 4. Afterwards, the results of this simulation are shown in Chapter 5. These results are primarily considering the performance regarding costs and service levels. Additionally, Chapter 6 describes a small extension of our research considering the computational tractability of the different methods in terms of computation times. Finally, our results are discussed in Chapter 7 and directions for future research are provided.

1.2 Problem Context

Many production processes are subject to capacity constraints. Therefore, order release decisions should be taken to satisfy future and current demand with timely produced stock against minimum costs. These decisions consider both the coordination of material and the allocation of available resource capacity. However, this capacity can be restrictive in a way that it is insufficient to satisfy all customer demand which results in lost consumer sales. An example of capacity restrictions is that raw-material and commodity suppliers might have a finite inventory of their goods and may have to apply rationing, which will result in a limited supply for production facilities downstream in the supply chain. Likewise, manufacturing facilities could suffer inadequate capacity for short-term production and therefore be unable to fulfill all downstream orders. Another source of capacity restrictions could be, for example, disruptions in upstream stock points. Apart from these restrictions on capacity, and the costs related to utilizing the companies resources, one should also consider the association between production planning, the layout type of the production system, and its organizational structure. In practice, there exist many different types of production systems, and these differences will all significantly affect the kind of lot-sizing model that should be applied in a specific situation (Buschkühl, Sahling, Helber, & Tempelmeier, 2010). If our supply chain system consists of different production units which have job-shop/flow-shop processes, then one can model these systems with the Multi-Level Capacitated Lot-Sizing Problem (MLCLSP). The MLCLSP is often used as the elementary concept underlying Manufacturing Resource Planning (MRP) modules. Compared to the formalized MLCLSP, standard MRP planning software will give priority to the demand side of the problem and overlook the capacity restrictions when dealing with order release decisions (Hopp & Spearman, 2000). Standard MRP is inadequate to solve problems in industrial practice due to its inability to create feasible capacity plannings.
An essential element of supply chain management is decision making regarding material coordination and resource releases in the supply chain. These decisions must be cost optimal and provide adequate service levels. This coordination problem is also referred to as Supply Chain Operations Planning (SCOP) (De Kok & Fransoo, 2003). We can define three key features of SCOP (Spitter, 2005). First, the supply chain has to be modeled as a network. For the production of an end item the raw material will undergo multiple transformation activities. These transformation activities depend on the stage of the supply chain. At earlier stages, raw materials are converted into specific parts. In later stages, these components are assembled into larger modules/intermediates which eventually constitute the end-items. Physical transformation processes transform the addressed items. These are, i.a., the manufacturing and assembly processes. But these activities can also be non-physical, for example, transportation between locations. The second feature of SCOP is material coordination and resource release decisions. The release of both material and resources is determined by the SCOP function, and there is explicit attention for the simultaneous coordination of these decisions in a multi-echelon, multi-item setting. Finally, SCOP seeks optimal cost/profit performance while satisfying service targets. The service level can be defined in numerous ways. Several service measures are defined in Silver, Pyke, and Peterson (1998). E.g., one could compute the fraction of demand that is fullfilled from stock the, also called the P_2-service level or fill rate, or define customer service to be the non-stock-out probability at the end of a replenishment cycle, which is called the P_1-service measure. SCOP performance can be measured by incorporating a rolling horizon schedule (De Kok & Fransoo, 2003). In the context of our problem, we will look at finite horizon scheduling of production and inventory decisions. In most industries, the planning horizon of the SCOP function boils down to about 12-14 months, and time buckets to aggregate demand and capacity are scaled to weeks. In subsequent periods the SCOP function determines a planning based upon demand forecasts. The planning is executed by releasing the appropriate materials and resources, but only for the current period. Afterward, demand forecasts will be updated by incorporating actual customer demand information, and a new production schedule is generated for future periods within the planning horizon while actual inventory levels and outstanding production orders are taken into account.
Uncertainty is an important aspect inherent to SCOP and an important consideration while

managing supply chain problems in general. Supply chain planning models should incorporate uncertainty. Otherwise, they will obtain worse results compared to methods that formalize this stochastic behaviour (Peidro, Mula, Poler, & Lario, 2009). There can be multiple sources of uncertainty in supply chain planning. A common source of uncertainty, and the main focus of our research, is customer demand. It represents unpredictability in demand by deviation from forecasts. Another source of variability is created by supplier deficiencies and disruptions upstream, which can be endogenized by multi-item multi-echelon inventory modeling. Lastly, uncertainty in process originates from unreliable production processes, e.g., machine breakdowns, unsatisfactory production yield, and batch rejections. However, for the sake of simplicity, we neglect the latter uncertainty construct.
Collectively the aforementioned aspects of the problem, i.e. lost sales, finite capacity, supply chain planning, and demand uncertainty, define our problem in its broadest sense. We conclude this context description by enumerating the specific characteristics of our problem.

- We consider the short-term optimization of the supply chain under consideration. Hence, we study the transient behaviour of the production-inventory system. We consider monthly time buckets and optimize a planning horizon of 6-12 months in a rolling-horizon scheme fashion, which is common in practice. Initial inventory levels have a significant impact on the transient behaviour of these systems.

- Demand which cannot be fulfilled from on-hand inventory is lost. Thus, our system considers lost sales. Our models focus on the minimization of lost sales and inventory holding costs. Hence, they implicitly maximize the expected profit. We elaborate on the latter statement in Subsection 1.3.1

- Demand is uncertain. We assume that forecasts exist which provide the actual values of the expected demand and the coefficient of variation of the end items. Furthermore, we have full information on the underlying probability distribution of the demand, i.e., the type of distribution and its moments are known.

- The underlying supply chain of the production-inventory system can be of a general structure.

Section 1.3.1 provides a formal and more concise definition of our problem. Figure 1.1 provides a small visual example of the type of problems that we consider in this thesis.

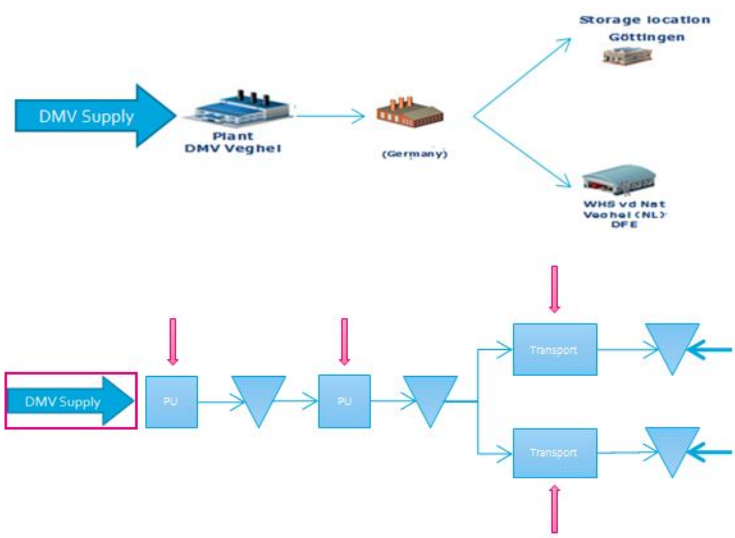


Figure 1.1: Example of SCOP problem under consideration

The chain under consideration comprises of several stock points and production units (PUs). The red arrows indicate capacity restrictions imposed on the PUs. In addition, the initial supply of material can also be limited. The methodology developed in this work will be applicable to general supply chain structures. Furthermore, a real-life case study will be executed based on the snippet provided above. In particular, the case study solves a production-inventory problem for one of the plants in the supply chain.

1.2.1 Business Relevance

In essence, the stated problem is omnipresent in industry. SCOP is an important concept for basically all types of supply chains. The problem that we will construct in the upcoming sections is capable of modeling real-life problems in an accurate way. We consider a general supply chain structure and allow for restrictions on the capacity of the PUs. Furthermore, as in practice, we focus on short-term planning according to a rolling-horizon scheme. To enable this flexibility/completeness in modeling our supply chain problem, we have to sacrifice the prospect of analytical expressions. The models that are developed in this research are linear programming (LP) and mixed integer programming (MIP) models. As such, these models have to be solved numerically by suitable algorithmic procedures. The developed algorithms can be implemented in contemporary advanced planning software (APS) modules. However, this dependency on numerical procedures may induce complications in the area of computational tractability when problem instances become relatively large.

1.3 Research Design

Note that this research is not following the regular standards of an Operations Management & Logistics thesis. The normal procedure adheres to the regulative cycle as described by Van Aken, Berends, and Van der Bij (2007). However, this study aims to develop new knowledge and is grounded in the Operations Research field. Therefore, this study is grounded in a different methodological framework. We will perform an axiomatic quantitative research as described by Bertrand and Fransoo (2002). A concise representation of the research phases is depicted in Figure 1.2.

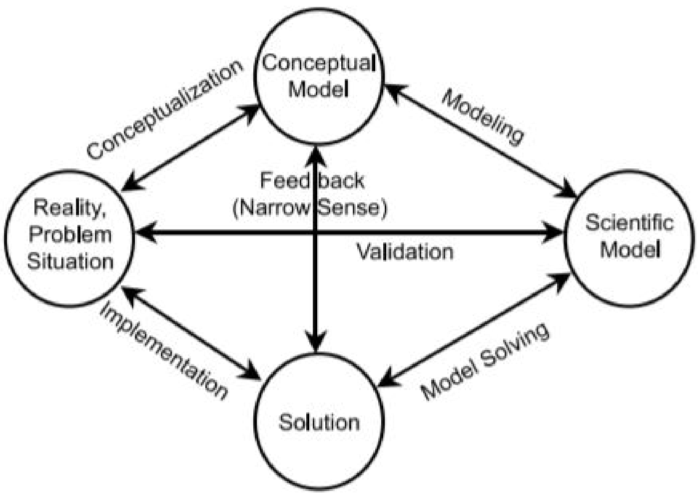


Figure 1.2: Operations Research phases (Mitroff et al., 1974)

To be more precise, the first phase consists of generating a conceptual model description. This is done in Section 1.2 and the literature study. These position the problem in a specific

business context and line of research. Next, we translate the conceptual model in a formal mathematical model. This scientific model enables mathematical analysis and/or simulation. The third phase consists of solving the mathematical model. However, our problem is highly intractable. Besides the fact that the deterministic MLCLSP on itself is already NP-hard when binary setup variables are included, the addition of stochastic parameters exacerbate the curse of dimensionality. The underlying state space is a tuple consisting of the time period, decision variable values, and uncertainty realizations. This results in a complex problem. We will derive (heuristic) techniques that capture the stochastic nature of the problem, and later on these methods will be tested by applying simulation on a factorial experimental design. Finally, we will compare the results and relate the findings to our research questions and the conceptual model.
Our study follows an experimental design of controlled instances. A set of instances will be constructed and thereafter analyzed in a simulation study. Several parameters of the problem are controlled. We use the aforementioned optimization models to solve distinct instances of the problem and compare several performance measures, such as average costs, aggregate service levels, and worst-case performance, using Monte Carlo simulation. This design should provide insights in the planning behaviour of the decision algorithms. Figure 1.3 briefly summarizes the setup of our research.

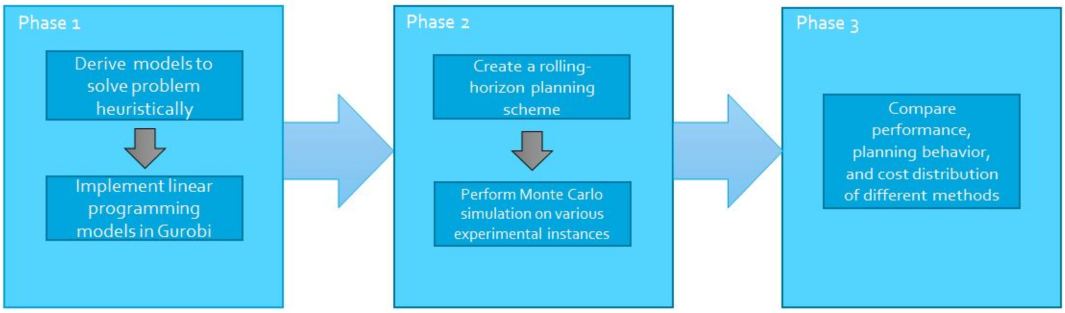


Figure 1.3: General setup of the study

1.3.1 Problem Statement

Let us first provide a generic formulation of the problem. First, let us consider the so-called inventory and lot-size (I&L) formulation of the MLCLSP provided in the work of Buschkühl et al. (2010). This portrays a deterministic planning problem which we use as a foundation for our model. This problem considers the production-inventory planning across a supply chain structure over a finite time horizon. We slightly diverge from this model in different ways. First of all, we consider a lost-sales environment instead of backordering. Furthermore, we incorporate the setup times in terms of the minimum order quantity (MOQ) concept. By introducing a MOQ, we can abstract from setup and carryover concepts by converting associated capacity consumption to a minimum lot-size. In industry, this technique is often used by planners on the goods flow control level to avoid the additional complexity that arises with scheduling decisions. This enables us to decouple the supply chain planning decisions on the tactical level from the detailed scheduling decisions on the operational level. Therefore, it enables us to carry the MLCLSP to a planning level similar to that of SCOP. Note that we use the term product, item, and stock-keeping unit (SKU) interchangeably. We introduce the following sets:

- $\mathcal{T}$: the set of discrete time periods in our planning horizon with elements ranging from $t = 1, ..., T$.
- $\mathcal{M}$: the set of resources used in the processes of the supply chain system, which have finite capacity. It consists of the elements $m = 1, ..., M$.

- $\mathcal{K}$: the set of SKUs ranging from $k = 1, ..., K$.
- $\mathcal{E} \subset \mathcal{K}$: the set of end items.
- $\mathcal{K}_m \subseteq \mathcal{K}$: a subset of items $k \in \mathcal{K}$ that need activities provided by resource m.
- $\mathcal{S}_k \subset \mathcal{K}$: a subset of items $i \in \mathcal{K}$ that require item k in their transformation activities.

Next, we define the input parameters of the mathematical programming (MP) model:

- d_{kt}: the random variable that represents the (external) customer demand for item k in period t.
- c_{mt}: input parameter that represents the capacity of resource m in time period t.
- a_{kj}: the internal demand for SKU k that originates from the requirement to produce one unit of product j.
- q_k^{min}: the minimum order quantity that should be produced when a setup for item k occurs.
- p_k: the processing time for producing one unit of item k.
- h_k: the holding cost of item k per unit that is not sold at the end of a time period.
- b_k: the lost-sales opportunity cost of item k per unit short at the end of a period.
- z_k: the planned lead time for activities regarding SKU k.

Finally, let us provide the decision variables of our model:

- Y_{kt}: a binary setup variable that determines whether item k will be produced in period t.
- Q_{kt}: the production quantity of item k in period t, which will be available at the start of period $t + z_k$.
- I_{kt}: the inventory on hand of SKU k at the end of period t.
- L_{kt}: the lost sales for SKU k at the end of period t.

We based the model on the I&L formulation of the MLCLSP (Buschkühl et al., 2010). However, the notation is slightly adapted to incorporate the fact that we are considering a stochastic environment. Furthermore, we introduce the MOQ concept to implicitly incorporate setup times as discussed above. Moreover, our model considers a lost-sales environment. Demand parameters are modeled as random variables to incorporate the aforementioned uncertainty. We first present the formal model and elaborate on the objective function and constraints afterwards. The model is defined as follows:

$$\min \quad \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}[h_k I_{kt} + b_k L_{kt}] \tag{1.1}$$

$$s.t. \quad I_{k,t-1} + Q_{k,t-z_k} - \sum_{j\in\mathcal{S}_k} a_{kj}Q_{jt} = I_{k,t} \qquad \forall k \in \mathcal{K}\backslash\mathcal{E},\ \forall t \in \mathcal{T} \tag{1.2}$$

$$I_{e,t-1} + L_{et} + Q_{e,t-z_e} = I_{e,t} + d_{et} \qquad \forall e \in \mathcal{E},\ \forall t \in \mathcal{T} \tag{1.3}$$

$$\sum_{k\in\mathcal{K}_m} p_k Q_{kt} \le c_{mt} \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1.4}$$

$$Q_{kt} \ge q_k^{min} Y_{kt} \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{1.5}$$

$$Q_{kt} \le (\sum_{m:k\in\mathcal{K}_m} c_{mt})Y_{kt} \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{1.6}$$

$$I_{k0} = I_{kT} = 0 \qquad \forall k \in \mathcal{K} \tag{1.7}$$

$$Q_{kt}, I_{kt}, L_{kt} \ge 0 \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{1.8}$$

$$Y_{kt} \in \{0,1\} \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{1.9}$$

For convenience, we will refer to the MOQ definition when mentioning this variant. This is the MLCLSP formulation which will function as a foundation for the experimental design in Sections 4.2 and 6.2. First of all, let us consider the objective function (1.1). This objective is equivalent to optimizing the profit in the present supply chain. In our formulation, this component is the sum of holding costs and lost-sales penalties. This equivalence comes about in the definition of the opportunity cost b_k. Let b_k equal the profit made on selling one unit of item k. When using an accrual accounting policy, we can compute the profit made on selling item k in period t by multiplying the selling price r_k and total amount of sales S_{kt} and thereafter subtracting the the cost of goods sold. In particular, we have that $\mathbb{E}[LS_{kt}] = \mathbb{E}[d_{kt}] - \mathbb{E}[S_{kt}]$ and the gross profit equals $(r_k - f_k) \cdot \mathbb{E}[S_{kt}] = b_k \cdot \mathbb{E}[S_{kt}]$, where f_k is the cost of producing one unit of k. Note that $b_k \cdot \mathbb{E}[LS_{kt}]$ is the lost profit for item k in period t. Furthermore, we subtract the inventory holding costs that represent the cost of capital required to carry the current inventory on hand. Hence, optimizing (1.1) is equivalent to optimizing profits whilst considering inventory investments. Constraints (1.2) and (1.3) are the inventory balance constraints. Different supply chain structures can be modeled, which are captured in the Bill-of-Material (BOM) given by the a_{kj} input parameters. Constraints (1.4) are the capacity constraints. These restrict the amount of activities that can be executed by resource m in time period t. Additionally, Constraints (1.5) restrict our model to always manufacture an MOQ when an item k is produced in time period t. Constraints (1.6) ensure that item k cannot be produced in period t when the binary setup variable Y_{kt} equals zero. Moreover, we made this bound as tight as possible to enhance the running time performance of the Branch-and-Bound (B&B) algorithm. Using the trivial big M bound enlarges the feasible space of the linear programming relaxation, which causes B&B to find integral solutions less efficiently. The ending and starting inventory is restricted by Constraints (1.7). In the standard model the value is set to zero. However, there could be scenarios were a certain amount of starting inventory is available, or were it is deemed necessary to build up stock for future periods which are not in the current planning horizon T. Finally, Constraints (1.8) and (1.9) restrict the decision variables to lie in the non-negative orthant and be an element of the set $\{0, 1\}$, respectively. For the first set of experiments we will not consider the binary setup decisions.
The stochastic components are already illustrated in this model, but these need a more rigorous definition. We model the uncertainty associated with the parameters in two different ways. We use probability spaces and uncertainty sets. The former is an ingredient for scenario generation in multi-stage stochastic programming algorithms and is a construct from probability theory. The latter is used in robust optimization methodology and takes a geometric approach to model uncertainty.
In the case of uncertainty in demand, the parameter d_{kt} is modeled as a random variable. Let $(\Omega_d, \mathcal{F}_d, \mathbb{P}_d)$ be the probability space encompassing the randomness in demand. Ω_d is

a sample space that represents all possible outcomes that could occur in the time course of the demand process for all items in the entire supply chain. $\mathcal{F}_d$ represents a σ-algebra of events, and $\mathbb{P}_d$ is the probability measure equipped with Ω_d. The demand is modeled as a random matrix $\mathbf{D} : \Omega_d \rightarrow \mathbb{R}^{K\times T}$. Define $\mathbb{P}(\mathbf{D} \in E) := \mathbb{P}(\{\omega \in \Omega_d : \mathbf{D}(\omega) \in E\}) \quad \forall E \in \mathcal{F}_d$, $\mathcal{F}_d = \sigma(\mathbf{D})$. This notation is used to abstract from underlying probability distributions and dependencies between random variables. Hence, it captures discrete, continuous, and mixtures of distributions. Furthermore, it accounts for dependent random variables. The probability spaces are used in the scenario generation process.
To use robust optimization techniques, one has to define uncertainty sets that capture the worst-case behaviour of the input parameters. We will use the budget of uncertainty method, which was proposed by Bertsimas and Sim (2004). In this model, the uncertainty set $\mathcal{U}$ takes on a polyhedral structure. Let us first define the set that reflects the uncertainty in demand. The assumption is made that d_{kt} is a random variable which is bounded and is symmetrical around a central value. This implies that realizations of this random variable lie in $[\mu_{kt} - \hat{d}_{kt}, \mu_{kt} + \hat{d}_{kt}]$. μ_{kt} is the expected value and $\hat{d}_{kt}$ is the assumed maximum deviation. Furthermore, define $\xi^d_{kt} = \frac{d_{kt}-\mu_{kt}}{\hat{d}_{kt}}$ to be the scaled deviation. This parameter falls in the interval $[-1, 1]$. Subsequently, we let the random variable conform to the affine function $d_{kt} = \mu_{kt} + \hat{d}_{kt} \cdot \xi_{kt}$ and define the uncertainty set to be

$$\mathcal{U}_d = \left\{ \mathbf{D} \in \mathbb{R}^{K\times T}_{+} \mid \xi^d_{kt} \in [-1, 1], \sum_{\tau=1}^{t} |\xi^d_{k\tau}| \leq \Gamma^d_{kt}, \forall k \in \mathcal{K}, t \in \mathcal{T} \right\} \tag{1.10}$$

This set is clearly a polyhedron. The Γ_{kt} parameter imposes an additional constraint on the values of coefficients. It induces that only a limited amount of parameters can take on their boundary values and places restrictions on the maximum deviations. This parameter is also called the uncertainty budget and it portrays the risk aversity of the decision maker. We can incorporate this uncertainty set in the optimization model as possible realizations of the input parameters and instantiate a min-max optimization model.

1.3.2 Scope and Assumptions

In this subsection we set the scope of our research. As the MLCLSP encompasses a variety of different modeling aspects and reflects different real-life planning and scheduling problems, we provide some indications of which type of problem this study solves and mention the associated assumptions.
First of all, we slightly remodeled the I&L formulation of Buschkühl et al. (2010) as discussed in Subsection 1.3.1. The MOQ concept has been introduced. This concept enables us to decouple the MLCLSP from detailed scheduling decisions that are made in the PUs. Hence, this enables us to position the MLCLSP on a similar planning level as the SCOP. The original I&L formulation and the variant incorporating setup carryovers incorporate detailed scheduling decisions regarding the setups of resources in the PUs. Therefore, these models are linked to operational decisions, but our research aims to abstract from these decisions and the MOQ concept enables us to do so. Hence, the main focus of our research lies on the goods flow control planning level. It can be argued that this variant of the MLCLSP is positioned higher on the planning hierarchy than the SCOP because it does not distinguish between material order releases and resource allocation decisions (De Kok & Fransoo, 2003).
Second of all, this study will only examine transient behaviour of the supply chain system. In industry, it turns out that most problems related to capacity allocation are induced by a capacity shortage in the short-term planning horizon, i.e, six to eight months in most industries. Therefore, this study will concentrate on planning decisions for a finite horizon T. As the decision making is done on the tactical level, the time buckets are scaled to months.
Third of all, we assume that the end-items in our MLCLSP model are the products situated in the Customer Order Decoupling Point (CODP) stock point. These SKUs do not have to be the actual final products that are delivered to the customer. The CODP is positioned in the supply

network such that orders can be manufactured Make-to-Stock within the requested customer lead time. Hence, our problem does not consider the placement of the CODP, nor does it treat detailed order scheduling at the PUs downstream of the CODP in the lower echelons of the supply chain.
There is no explicit treatment of forecasting issues on the goods flow control planning level. This implies that we do not utilize different forecasting methods to capture the demand variability. Instead, we assume complete knowledge about the probability space of the demand and suitable bounds for the uncertainty set. The probability space is used to generate scenarios for the multi-stage stochastic programming procedures, and the uncertainty set is used for the robust optimization techniques. Also, the bounds on the demand realizations should be symmetric around a central value.
Furthermore, the system that we model has complete flexibility. This implies that capacity allocation and inventory flow decisions can be changed in each time period. Additionally, this is done in a rolling-horizon fashion. Each period a demand realization occurs and the optimization methods decide for future periods given the past.
Finally, we neglect the interference of the product lifecycles of the end items on the planning process. This assumption is quite plausible as we consider a short-term planning horizon. However, note that some industries, e.g. the high-tech industry, have relatively short cycles, and hence the associated planning processes should incorporate product introductions and phase-outs.

1.3.3 Optimization Techniques

This section provides a short introduction to the different optimization approaches. These methods include:

1. Deterministic LP-based rolling-horizon planning. This is our benchmark optimization model. Many planners in practice claim that solving these LP problems generates the true optimal result. However, our world is inherently uncertain, and hence it is necessary to model certain input parameters as stochastic processes. This optimization model resolves the situation were the demand is assumed to be known in advance. In our variant of the model, the demand parameter is set to the expected value.

2. Safety stock heuristic. This is an enhanced reformulation of the benchmark LP model. This model incorporates the uncertainty in demand by targeting a required safety stock level. However, this procedure generates these safety stock by using the theory of Diks and De Kok (1998). Unfortunately, these newsvendor type of equations do not hold in the lost-sales, capacitated planning problem that we are considering. Nevertheless, it provides us with a heuristic that incorporates the uncertainty in demand.

3. Robust optimization. This method approaches modeling the uncertainty of demand in a different way. The uncertainty of the demand is captured by an uncertainty set, and the worst-case performance of the system is minimized over this entire set. Hence, the uncertainty is captured in a geometric fashion. We define the worst-case performance as the maximum cost incurred during the Monte Carlo simulation.

4. Sample average approximation. This is an approximate method for solving the exact multi-stage stochastic programming model that can be derived from the model given in Subsection 1.3.1.

5. Distributionally robust optimization. This method optimizes the problem with respect to the worst-case expected performance. Hence, this approach positions itself in an area between the geometric approach of RO and the stochastic solution method SAA.

1.3.4 Research Questions

Our study will address four main research questions. In this subsection we discuss them in more detail and relate them to the gaps in literature found during the literature study, which are stated in Chapter 2. The problem is specified in Subsection 1.3.1. The experimental design to answer these questions is given in Section 4.2. The first question is:

> *"Which solution technique provides the best performance when solving distinct instances of our problem?"*

The solution techniques that will be compared are explained in more detail in Chapter 3. Performance is an encapsulation for different metrics. This study will compare average total costs, variances of these costs, and worst-case supply chain costs. The distinct instances refer to the experimental instances that will be described in Section 4.2. We will evaluate several different values for specific input parameters. To our knowledge, no variant of the MLCLSP has ever been subject to a study comparing the methodologies described in Chapter 3. Curcio, Amorim, Zhang, and Almada-Lobo (2018) perform a similar study on the General Lot-Sizing and Scheduling Problem (GLSP) and mention the extension to multi-level problems as future research.
Our second research question relates to the actual decisions that are generated by the different optimization techniques. This question reads:

> *"What differences can be observed in system behaviour when comparing the decision algorithms?"*

Again we let the solution algorithms resolve the experimental instances. However, this time we focus on the underlying planning behavior of the system. I.e., we look at the propagation of inventory through the supply chain and capacity allocation of items on the different resources. The goal is to provide insights in the underlying behaviour of the different decision-making techniques. Analogously to the comparison of the LP-based concepts and the SBS policies in the work of De Kok and Fransoo (2003), our research compares the rolling-horizon LP-based methodology with techniques which originated from stochastic decision theory and robust optimization. This study explores the difference in their underlying decision structures in the context of our problem.
The third question expands on our original design of experiments. Note that we omit the binary setup decision variable in the first part of our research. The additional challenges that arise by introducing the MOQ concept is the topic for the third research question:

> *"Which methods are acceptable regarding computation times when incorporating setup decision variables into the MOQ formulation of the MLCLSP?"*

In general, the addition of setup decisions results in the problem becoming NP-hard. Therefore, some of the techniques that are applied will become computationally intractable. Hence, different relaxation assumptions must be applied and tested with regard to solution quality and running-time performance. Present-day research avoids this problem by relaxing future setup decisions (Curcio et al., 2018; Alem, Curcio, Amorim, & Almada-Lobo, 2018), especially when applying robust optimization techniques (Yanıkoglu, Gorissen, & Den Hertog, 2018).
Finally, we are interested in proving our methodology in real-world applications. Hence, our final research question is:

> *"What flexibility exists in the configuration of our methodology to solve real-life business cases?"*

Appendix D describes a real-life business case. The techniques that are developed in earlier sections are applied to this case to provide a proof of concept.

1.4 Contributions to Literature

Our contribution is twofold. First of all, we emphasize the relevance of our research for supply chain planning problems in industry. In essence, the problem defined is the stochastic ML-CLSP (S-MLCLSP), which is a manifestation of the SCOP construct. Note that no variant of the S-MLCLSP has ever been subject to a study comparing the methodologies used in this thesis. Solving this problem is extremely involved, and there are no guarantees for any kind of mathematical optimality for the proposed solutions. There exists an abundance of research which considers tractable stochastic models that neglect capacity constraints and can only represent supply chains with restricted complexity. In order to deal with less idealized supply chains and capacity restrictions, one has to rely on MP-based methods in a rolling horizon scheduling context. Our research provides several appropriate heuristic methods that are able to resolve the S-MLCLSP. However, note that these mathematical programming models cannot be validated as representing real-life systems. The stochastic process formulation can be perceived as empirically valid for the true system (De Kok, 2018). In particular, the SAA method approximates this reality.
Secondly, all production-inventory models discussed in previous research assume backordering. To our knowledge, we are the first to consider lost-sales inventory models for robust optimization approaches. In the case of backordering, the models discusses possess the helpful property that the cumulative net stock follows a trivial relation. Unfortunately, lost-sales models lose this property due to truncation of the net stock's domain at zero. As such, the derivations for the lost-sales RO and DRO models provided in Chapter 3 are first of a kind. These derivations use fundamental techniques for these types of models.

Chapter 2

Literature Study

This chapter provides an overview of relevant literature regarding the S-MLCLSP. First of all, we elaborate on the connection between the S-MLCLSP and the concept of SCOP. Thereafter, Section 2.2 discusses the single-echelon building blocks for the MLCLSP and their solution methodologies: the Economic Lot-Sizing Problem (ELSP). Subsequently, Section 2.3 discusses the deterministic instance of our problem, which we later on expand to the stochastic variant in Section 2.4. Finally, we discuss the existing research gaps in Section 2.5.

2.1 Connection to Supply Chain Operations Planning

This section of the literature study will be dedicated to defining the SCOP problem and the theory related to it. The majority of this section is based on the work of De Kok and Fransoo (2003).

The SCOP problem
In their chapter of the handbook, De Kok and Fransoo (2003) define SCOP as the coordination of material and the resource release decisions in the supply chain such that service levels are satisfied with minimum costs/maximum profit. Hence, SCOP is related to the coordination of good flows and the allocation of capacity in a supply network. SCOP distinguishes itself from other supply chain management decision processes in different ways:

1. Material and resource constraints are considered at the same time.
2. The material release decisions are taken in a multi-level, multi-item, longitudinal decision setting, and therefore incorporates operational decision making over time.
3. SCOP relates to both tactical and operational planning, and thereby transforms mid-term planning into short-term operational decisions.
4. SCOP models incorporate demand and capacity uncertainty due to the quasi-focus on operational coordination.

The SCOP problem incorporates the coordination of activities in the entire supply chain. Products are interrelated based on the supply chain structure and all intermediate products are transformed into other intermediates or end-items. Note that these days the original equipment manufacturers and the first and second tier suppliers are all part of the aforementioned supply network. The SCOP problem also involves different functional units within organizations, such as the sales and marketing department or the manufacturing and warehousing functions. In the remainder of this section, a centralized objective function is assumed. Furthermore, we assume that all information is shared across the entire supply chain.

The SCOP function

The SCOP function coordinates all activities in the supply chain. It makes decisions on the material and resource releases and supports all steps in transforming specific input items into end-items using the available capacity and applying multiple transformation activities. A small list of exemplary transformation activities:

1. Manufacturing, these activities will physically transform the input items into intermediates or end-items.

2. Transportation, moving physical outputs between different locations.

3. Planning, the administrative tasks required for the other types of activities to be enabled.

In general there exist two types of relationships between physical transformation activities. First, the input of one transformation activity is the output of another activity. Secondly, a transformation activity has one or multiple resources in common with other activities. The first relationship is translated into the BOM. The second aspect is structured by defining the Bill-of-Process.

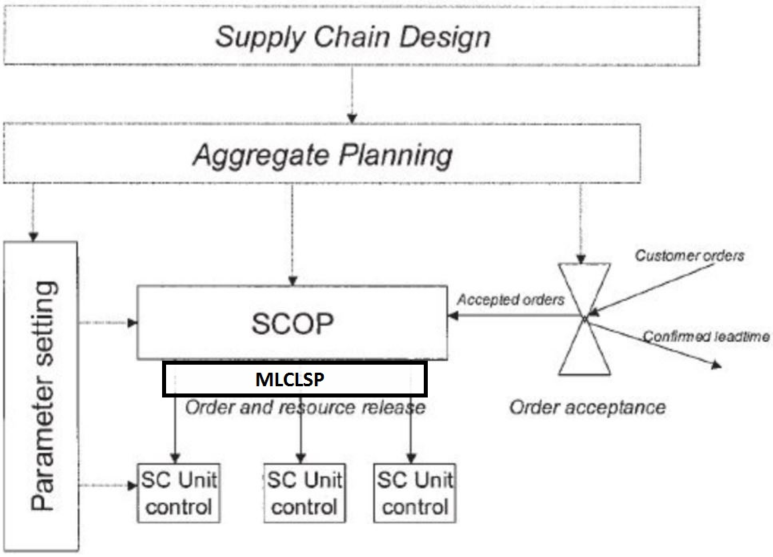


Figure 2.1: Positioning SCOP in Hierarchical Planning Framework; edited from (De Kok & Fransoo, 2003)

Conceptual positioning of SCOP in the hierarchical planning framework

Figure 2.1 displays the positioning of SCOP in the hierarchical planning framework described by Bertrand, Wortmann, and Wijngaard (1990). SCOP can be positioned above the functional unit control functions. These PU functions guarantee the planned lead times of specific processes/activities in the supply network in accordance to predefined benchmarks. The order acceptance function is introduced to manage the total workload that enters the supply chain system. Additionally, it affirms the lead time towards the customer. The SCOP provides order and resource release decisions as input to the supply chain units, which perform the detailed scheduling of the orders and are assigned with the goal of minimizing internal costs while conforming to the planned lead-time requirements. The SCOP controls the total amount of workload it releases to the PUs by using an anticipation function. A parameter setting function coordinates the safety stock settings, planned lead times, and total workload.

Higher level decisions are related to long-term aggregate planning and supply chain design. The MLCLSP is also positioned in the framework displayed by Figure 2.1. When solving the prevalent formulations of the MLCLSP, different detailed production-scheduling decisions are considered. Therefore, we position the MLCLSP on a slightly more operational level. However, as the SCOP considers more elaborate decision making regarding the resource and order release decisions over time, one could discuss the previous statement.

2.2 Economic Lot-Sizing Problems

In this section, multiple variants of the ELSP are reviewed. Lot-sizing is extensively studied in Operations Research. This section only discusses the single-level problem. This elementary problem is a building block for the more complicated multi-level variant.

2.2.1 Variants of Lot Sizing Problems

Let us discuss a collection of different lot-sizing and scheduling problems. Our main concern will be deterministic, single-level lot-sizing problems with dynamic demand. Particularly, we will discuss the Capacitated Lot-Sizing Problem (CLSP). Preliminary, five variants of this problem will be discussed.

First of all, lets us consider the ELSP as discussed in Gallego and Shaw (1997). This problem is characterized by a single-level, multiple end items, and stationary demand. When incorporating restrictions on the capacity of the system, it turns out that the ELSP problem is NP-hard. It is common to model time as a continuous parameter and set the planning horizon T to infinity. The Discrete Lot-Sizing Problem is a variant of the regular ELSP (Bruggeman & Jahnke, 2000). It differs from ELSP by considering discrete time buckets. These time buckets are characterized as small. As such, the production decisions per period are termed all-or-nothing solutions. The planning solution will produce one product per period using full capacity. This problem relates to the Continuous Setup Lot-Sizing Problem by incorporating discrete time periods (Drexl & Kimms, 1997). However, the latter problem has a better resemblance to reality. It departs from the all-or-nothing assumption and enables the model to decide upon production quantities in each period. The Continuous Setup Lot-Sizing Problem considers larger time buckets than the Discrete Lot-Sizing Problem but still assumes that only one item can be produced each planning period.

The Proportional Lot-sizing and Scheduling Problem extends on the latter models by producing a second item in a planning period when there exists leftover capacity (Drexl & Haasse, 1995). The GLSP generalizes this to the production of an arbitrary number of items on a single machine with capacity restrictions (Fleischmann & Meyr, 1997).

To conclude this section, a formal definition of the CLSP will be stated according to the definitions given in Karimi, Fatemi Ghomi, and Wilson (2003). Consider a single-level system with a finite planning horizon T, a deterministic dynamic demand d_{it} for item i in planning period t without backordering, capacity restrictions R_{it}, production costs C_{it}, setup costs S_{it}, resource consumption in units a_i for product i, and holding costs h_{it} for item i at the end of period t. Furthermore, define the following decision variables: the lot-size X_{it} of item i in period t, the inventory I_{it} of product i at the end of t, and the binary variable Y_{it} to determine whether i is produced in period t. The following mixed integer programming model can be stated:

$$\min \quad \sum_{i=1}^{n}\sum_{t=1}^{T}(S_{it}Y_{it} + C_{it}X_{it} + h_{it}I_{it}) \tag{2.1}$$

$$s.t. \quad X_{it} + I_{i,t-1} - I_{it} = d_{it} \qquad i = 1, ..., n; \; t = 1, ..., T \tag{2.2}$$

$$\sum_{i=1}^{n} a_i X_{it} \leq R_t \qquad t = 1, ..., T \tag{2.3}$$

$$X_{it} \leq MY_{it} \qquad i = 1, ..., n; \; t = 1, ..., T \tag{2.4}$$

$$X_{it}, I_{it} \geq 0 \qquad i = 1, ..., n; \; t = 1, ..., T \tag{2.5}$$

$$Y_{it} \in \{0, 1\} \qquad i = 1, ..., n; \; t = 1, ..., T \tag{2.6}$$

Our objective function concerns the minimization of the total costs. These costs consist of setup costs, production costs, and holding costs. Furthermore, the model contains the usual inventory balance and capacity constraints. This problem is proven to lie in the space of NP-hard problems as shown by Florian, Lenstra, and Rinnooy Kan (1980) and Bitran and Yanasse (1982), and even constructing a feasible solution to this problem is NP-complete (Maes & McClain, 1991).

2.2.2 Solution Methods for Determenistic Capacitated Problem

First of all, let's consider some solution methodologies for the uncapacitated lot-sizing problem as a baseline. The Economic Order Quantity (EOQ) as introduced by Harris (1913) is one of the most well-known solutions to a trivial variant of the ELSP. The EOQ model assumes a constant demand rate and an infinite planning horizon. The popular EOQ formula is used in practice for determining the optimal balance between setup costs and holding costs by determining a suitable lot-size. In the 1950's, Wagner and Whitin (1958) developed an exact solution method for the uncapacitated lot-sizing problem with a single item and deterministic dynamic demand. The algorithm used a dynamic programming approach. Following this research, multiple authors have built upon this method and created new algorithms with increased performance or generalizability. In the paper of Wagelmans, Van Hoesel, and Kolen (1992) a solution algorithm is proposed that solves the problem of Wagner and Whitin in linear time, and Federgruen and Tzur (1991) implemented an algorithm that can solve more general versions of the dynamic lot-sizing problem with $\mathcal{O}(n \log n)$ running time complexity.

Exact approaches

Considering the NP-hardness property of the CLSP, it is difficult to determine provable optimal solutions. There exist 3 different approaches for solving the mathematical programming model of the CLSP exactly. The first method is commonly used with these type of mathematical programming problems. We use a mathematical programming solver like CPLEX to solve the MIP formulation of the problem using Branch-and-Bound techniques.

A second approach is introduced by Barany and Van Roy (1984) and Leung, Magnanti, and Vachani (1989). Their method uses the creation of cutting planes. This boils down to adding stronger valid inequalities to the list of mathematical programming constraints. These cuts are determined by considering the single-item uncapacitated variant of the problem and reformulating the Wagner-Whitin solution schedules into linear constraints. These cuts adjust the feasible solution space of the LP-relaxation of the CLSP. This new space approximates the convex hull of the MIP formulation of the CLSP. The approximation will get better by adding more cutting planes. The new problem will be solved by the Branch-and-Bound procedures discussed in the former approach.

A third approach is given by Eppen and Martin (1987). Their approach transforms the regular CLSP formulation into a graph-type problem. This new formulation has grown in size as it needs a bigger number of variables and more constraints, but exceeds the traditional formulation by providing a tighter LP-relaxation bound. This relaxation bound can be used to prove optimality of a solution generated by branch and bound. The solution time will be reduced due to this tightening as the reformulation's solution space more tightly resembles the convex hull of the original MIP.

Problem-specific heuristics

The methods presented in the previous section require a tremendous amount of computational effort when considering large instances of the problem. Most of this effort lies in proving optimality of the solution. Heuristics are general methods that explore specific structures in the problem to provide *good* solutions. This section discusses some of the most frequently used types of heuristics for solving the CLSP. In the paper of Maes and Van Wassenhove (1988), heuristics are divided into two groups: the period-by-period heuristics and improvement heuristics.

The former heuristics consider each period separately. The requested quantities for all items are produced for a specific period and the leftover capacity will be assigned to demand in future periods. For the allocation of this excess capacity to products, a priority scheme should be constructed. A variety of different heuristics can be used to create the product priorities, e.g., part-period balancing or the lot-sizing heuristic of Silver and Meal (1973). In their paper, Maes and Van Wassenhove (1986) determine the required production quantities for a specific period. They calculate the remaining capacity and subsequently start allocating remaining capacity to future demand. The order in which items are checked for allocation is determined lexicographical. A heuristic method, such as Silver-Meal or least-unit cost, is applied to determine the demands that will be added to the production schedule. When using ABC rules to determine the allocation of production to demand, the extra cost that has to be payed is on average 3.64% more than the best solution looking at high time-between-order problems (Maes & Van Wassenhove, 1988). However, the computation time is approximately 100 times smaller when using this heuristic instead of branch and bound to solve the problem exactly. A variant of this type of heuristics called item-by-item is stated by Kirca and Kokten (1994). This method selects items that are not yet part of the current schedule and creates a planning for these items over all remaining periods. For selection of the mentioned items, the one-item algorithm is used. This algorithm uses the EOQ formula. This enables decomposition of the n-item problem into n different single-item problems. Thereafter, capacity is adjusted for the entire planning horizon. Their approach shows significant improvements over the period-by-period heuristics. In particular, the item-by-item approach outperformed the other approaches in 92.5% of the test cases and needs approximately 30 times less CPU seconds. Only for problem sets with many items and few periods the performance of both types of heuristics was similar.

The improvement heuristics consist of three steps. First of all, they will provide an initial solution by solving the uncapacitated lot-sizing variant of the problem. Subsequently, some additional constraints are applied to ensure feasibility of the solution. Finally, an additional improvement step is made to increase the value of the found solution. An example of this kind of heuristic is the algorithm created by Karni and Roll (1982). In the first step, the Wagner-Whitin solution of the uncapacitated version is generated. This provides us with the initial solution. The authors define ten types of shifts for allocating demand to different production periods. These shifts are then used in the second and third step to first create a feasible solution and thereafter reduce the costs of this solution. On average, the improvement approach costed 4.1% more than the optimal solution (again for high time-between-order problems), but this approach only needed two-thirds of the computation time that the ABC period-by-period approach requires (Maes & Van Wassenhove, 1988). For low time-between-order cases the differences were less explicit.

Mathematical programming heuristics

The heuristics in the previous section are tailor made for the CLSP. This section discusses more general heuristic procedures based on mathematical programming. These heuristics can be modified to different kind of problems and in general generate better quality solutions than the problem specific heuristics. Furthermore, the mathematical programming heuristics provide lower bounds on the optimal solution and therefore create better insight into the quality of the found solution. However, these heuristics will be less comprehensible for practitioners, and when they are applied on real-life cases the models might become intractable. In this class there are three main types of heuristics: relaxation heuristics, B&B heuristics, and column generation heuristics.

The relaxation-based heuristics use Lagrangian relaxation on the capacity constraints. Due to this relaxation, the CLSP decomposes into n single-item problems without capacity restrictions. The algorithm of Thizy and Van Wassenhove (1985) uses this relaxation to decompose the problem, and they subsequently solve the problem using the Wagner-Whitin algorithm. Thereafter, they use the subgradient algorithm to optimize the dual problem and generate the best lower bound on the solution. The value of the setup variables is given by the solution of the relaxed problem and fixing these will result in a transportation problem. Solving this transportation problem generates an upper bound for the CLSP. The solution procedure keeps iterating until both bounds are equal or the maximum number of iterations is reached. Maes and Van Wassenhove (1988) showed that the relaxation-based heuristic was outperformed by the ABC period-by-period heuristic, except for the cases were utilization of the resources is low.

As regular branch and bound requires a lot of computation time, some heuristic procedures are constructed using this concept. An elementary example is the method provided by Armentano, Franca, and De Toledo (1999). The authors translate the CLSP into a minimum costs network flow problem. They create a solution to the flow problem by applying implicit enumeration.

The column generation heuristics are usually combined with set partitioning procedures. An exemplary method is given by Cattrysse, Maes, and Van Wassenhove (1990). In this method, some initial schedules are constructed with several heuristics. Subsequently, a selection is made by solving the LP relaxation of the set partitioning problem, hence integrating the column generation. Finally, heuristics are applied to create an integral solution. However, a major deficiency of this method is that under tight capacity constraints and a relatively small number of periods it may not provide feasible solutions. The experimental design of Cattrysse et al. (1990) was similar to that of both Maes and Van Wassenhove (1988) and Karni and Roll (1982). The former's best heuristic performed two to five percent better than the well-known common sense heuristics. Furthermore, the best approach of Cattrysse et al. (1990) was faster than the other heuristics that were discussed.

2.3 Deterministic Multi-Level Capacitated Lot-Sizing Problems

This section expands the model provided in Subsection 2.2.1 by adding multiple levels to the system. To enable modeling a supply chain, the suggested formulation should incorporate the parent-component relationships that exist in a supply chain network. Billington, McClain, and Thomas (1983) introduced the MLCLSP under a different name. They described their problem as capacity constrained multi-stage production scheduling, and the authors explained that their problem originates from the demerits of MRP systems. These shortcomings include the incompetence of MRP to cope with uncertain lead times, restricted capacity at facilities, uncertainty in demand, irregular yield, and changes in production scheduling for the end-items. The paper describes a linear and a mixed integer programming formulations for the problem. In their formulation, the production lead times are considered endogenous variables. I.e., the lead times are functions of capacity utilization. The authors introduce a minimum lead time that imposes the planning of production at least an equivalent number of periods ahead. If the capacity constraint turns out to be binding, then the production starts earlier than this minimum lead time. In this case, the production pass through inventory. Furthermore, the paper introduces product structure compression to make real-life cases tractable whilst still providing the optimal solution.

In their chapter, De Kok and Fransoo (2003) emphasize the fact that SCOP is a construct strongly related to the MLCLSP. The MLCLSP can be applied to different levels of the planning hierarchy. In our case, we will use the MLCLSP on the SCOP planning level for the purpose of optimizing tactical level production planning and supply chain planning. Note that the MLCLSP formulation does not treat release quantities and production quantities as separate variables. Therefore, we place the MLCLSP on a higher planning level, which determines aggregate quantities which are disaggregated in operational planning levels. Finally, note that the SCOP model formulation does not consider setup variables and lot-size related constraints. On the contrary, the MLCLSP formulation does incorporate these aspects, and therefore this model will be better applicable to the production/supply chain context that this study considers.

Erenguc, Simpson, and Vakharia (1999) provide an extensive review of the integration of production planning and distribution planning in supply chains. The authors decompose the supply chain in three different stages. The first stage considers the suppliers of raw materials and/or services to the manufacturing plants. The review places a narrower focus on this section of the supply chain because all operational decisions in downstream nodes are directly affected by the adjacent suppliers. Planning decisions in this stage include determining supplier criteria, supplier portfolio management, the type of relationship with the supplier, and the inventory control with respect to the supplier's shipments. The second stage consists of the manufacturing plants which transform the input acquired from the suppliers into the end-items. The essence lies in modeling the setup structures at the plants and the flow of material between them. The last stage considers the distribution network of the supply chain. This stage can consists of different entities, e.g., customers, wholesalers, regional/local distribution centra, etc. This stage considers decisions regarding the configuration of the distribution network, allocation of product demand to distribution centers, and the inventory control at each location. Erenguc et al. (1999) delineates methodology to optimize these supply chain stages. Ultimately, the authors try to integrate the planning models and the resulting problems become tremendously complex. Another important aspect of the analysis is the relationship between these different supply stages. If decision making is centralized and under control of a parent company, then information sharing and global optimization is achievable. In the case of other types of relationships, the different firms can decide to optimize locally and research should adjust to a game theoretical perspective.

A variety of lot-sizing models can be constructed from the MLCLSP. One could omit the dependent demand summation in Constraints (1.2) of the I&L formulation. This will result in the MLCLSP dissolving into M CLSPs. CLSP is discussed extensively in Section 2.2 of this literature study. The methods discussed in Subsection 2.2.2 can now be applied to solve the M single-level problems. The Multi-Level Uncapacitated Lot-Sizing Problem is a variant discussed by Robert Jacobs and Khumawala (1982). It can simply be derived from the MLCLSP when the capacity constraints are omitted. Other variants include the reduction to the LP formulation by fixing the setup pattern (Kuik & Salomon, 1990), or introducing auxiliary variables for backordering and overtime to create feasible solutions to the MLCLSP problem by introducing additional slack (Millar & Yang, 1994).

A final variant is based on the work of Suerie and Stadtler (2003). They formulate the Capacitated Lot-Sizing Problem with Linked lot sizes. This extension incorporates setup carryovers between planning periods by adding auxiliary constraints and variables to the capacitated lot-sizing model. The paper also translates this extension and formalizes the Multi-Level Capacitated Lot-Sizing Problem with Linked lot sizes (MLCLSPL).

2.3.1 Solution Approaches

In the remainder, different solution approaches for solving the deterministic MLCLSP are reviewed. This section conforms to the solution classification used by Buschkühl et al. (2010).

Five general groups of algorithms are considered. A couple of them function similar to the ones discussed in Subsection 2.2.2. The following classes are discussed: mathematical programming approaches, Lagrangian heuristics, decomposition and aggregation approaches, problem-specific greedy heuristics, and metaheuristics.

Mathematical programming approaches

The MP approaches are similar to the ones discussed in Subsection 2.2.2. In this section, we differentiate between heuristics based on MP and the exact approaches providing provable optimal solutions. The former approach searches a *good* solution in a truncated search space. It restricts computation time and only guarantees feasibility of the solution. The latter approach generates the optimal solution neglecting the usage of time and computer memory.

A common approach to solve MP problems is B&B which is already described in Subsection 2.2.2. Billington et al. (1983) combined these methods with Lagrangian relaxation. They apply a series of relaxations on the different items in the problem. The solution method has two distinct phases. In the first phase, the dual problem is solved and reduced costs are determined. In the follow-up phase, the primal problem is solved with these cost parameters.

An alternative method is problem reformulation. Tempelmeier and Helber (1994) created a method to convert the MLCLSP into a Shortest Route Problem (SRP). Subsequent research has dedicated itself on improving this methodology by improving on computation time. This was achieved by reducing non-negative coefficients in the corresponding LP matrices (Stadtler, 1997). This author also describes an approach to reformulating the MLCLSP into a Simple Plant Location Problem (SPLP) (Stadtler, 1996). Both approaches result in the same number of variables and have a similar objective function. However, the latter approach needs an extra $KT(\frac{1}{2}(T+1)-1)$ constraints when constructing the LP relaxation.

Adding additional inequalities to reduce the solution space is another suitable MP-based approach. This type of method is usually combined with B&B, and the approaches differ on the order of generating cuts and executing the Branch-and-Bound procedure. If the valid inequalities are added to the model before executing B&B, then the procedure is called Cut-and-Branch. In the reverse case, the algorithm is called Branch-and-Cut. Suerie and Stadtler (2003) apply both of these methods to the MLCLSPL. The cuts are generated by reformulating the setup carryover constraints. Furthermore, pre-processing is applied. Cuts are generated that tighten the existing solution space by exposing the relationships between inventory and setups, and the link between the production of a single item and total resource capacity.

The mentioned approaches can solve the problem to optimality when exempted from time restrictions. Alternatives are fix-and-relax heuristics and rounding heuristics. The former will fix a couple of binary decision variables in the original problem. The setups will be decision variables in the current period of the schedule and are fixed in earlier periods. All following periods have relaxed setup variables. Alternative approaches are developed that consider overlapping planning windows (Dillenberger, Escudero, Wollensak, & Zhang, 1994; Stadtler, 2003). Fixing the setup variables results in the original MIP problem reducing to a more tractable LP. The overlapping planning windows are modeled by an internally rolling schedule. A lot-sizing window is generated for each step in the internal rolling schedule, and the setup decisions are only made for the time periods that are considered in the current window (Stadtler, 2003). The method of Stadtler (2003) is compared to the work of Tempelmeier and Derstroff (1996), where the latter introduced a test bed that is commonly used for the MLCLSP and introduced a relaxation approach which we will discuss in the next paragraph. The method of Stadtler (2003) outperforms the heuristic of Tempelmeier and Derstroff (1996). The former achieved a better solution quality for basically all parameter combinations that were present in the test bed. Furthermore, the overall performance of this method was relatively stable. The performance gap between the heuristic of Stadtler (2003) and the optimal solution ranged between 0.90%-4.10%. CPU times were set to several different time limits for the method of Stadtler (2003). This method consumes a lot more computational time due to its complexity, but these times could be greatly reduced when utilizing the overlap of the planning windows and the internal structure of present-day MIP solvers.

Rounding heuristics transform the solution of the LP relaxation by rounding the fractional binary variables. In general, this results in infeasible solutions as capacity constraints may be violated. Rounding heuristics could be used for solving the SPLP or SRP formulation of the MLCLSP. For this purpose, the setup variables with the highest values can be rounded to one, and the amount of variables that are fixed can be subjected to a specific threshold (Maes & McClain, 1991; Alfieri, Brandimarte, & D'Orazio, 2002).

Lagrangian heuristics

These type of heuristics are based on Lagrangian relaxation. Lagrangian relaxation is a generic optimization technique that is applied to a plethora of different mathematical problems. Constraints that create interdependence are relaxed, substituted into the objective function, and weighted with Lagrange multipliers. Such constraints include the capacity constraints per resource as these create item dependencies and the inventory balance equations for a similar reason. When relaxation is applied on both sets of constraints the problem reduces to K uncapacitated single-level problems. Solving the relaxed problem provides a lower bound on the optimal solution.

A sophisticated solution algorithm is provided by Tempelmeier and Derstroff (1996). They apply the relaxations discussed above and complement them by low-level processing procedures. The authors solve the resulting single-level single-item problems in a specific order of the items that is determined by a detailed code. They consider a subset of items simultaneously to cover the multi-level nature of the original problem. Another approach is given in the paper of Özdamar and Barbarosoglu (2000). These authors created two different relaxation heuristics. One only relaxes capacity constraints, and the other relaxes both inventory balance and capacity restriction. By transferring production capacity they generate feasible solutions, and afterwards they apply simulated annealing to improve the generated solution. Özdamar and Barbarosoglu (2000) compare their methodology to that of Tempelmeier and Derstroff (1996). In the test bed used by Özdamar and Barbarosoglu (2000), it turns out that the heuristic of Tempelmeier and Derstroff (1996) had an average deviation from optimality of 1.39%. However, the latter heuristic was outperformed by the approaches presented by Özdamar and Barbarosoglu (2000). The heuristic of Tempelmeier and Derstroff (1996) had a slightly lower CPU time for the classes of problems that were tested. Problems with 10 items, 4 time periods, and 3 resources, were solved in approximately 1.5 seconds on average.

Decomposition and aggregation approaches

Decomposition of a problem transforms it into a collection of independent subproblems. These subproblems are generally more easy to handle. In the context of the MLCLSP, one can decompose the problem based on the products or on the time. In the former, the capacity constraints are omitted in the subproblems. For the latter, the problem is solved for each period separately and mainly in rolling schedule context.

Item-based decomposition is applied in the papers of Sambasivan and Schmidt (2002) and Tempelmeier and Helber (1994). The first paper discusses the decomposition of a multi-plant CLSP with dependencies into single-level uncapacitated problems. By using production smoothing the capacity restrictions are guaranteed, and the single-level problems are solved efficiently by translating them into SRPs. The second paper tackles the MLCLSP by providing two methods to decompose the main problem into CLSPs, and introducing two reduced cost modification procedures. A typical period-based approach is given by Bourjolly, Ding, Gopalakrishnan, Gramani, and Mohan (2001). They consider a rolling horizon situation. The subproblems, which are decomposed by considering accruing periods, are solved recursively by using the optimal schedule from the preceding periods. This approach can be adopted conveniently to allow for setup carryovers.

Aggregation approaches abstract from detailed planning decisions in the beginning. Later on, these methods will dissect the aggregate solution to make lower level decisions. In their paper, Özdamar and Bozyel (2000) introduce a novel planning method that aggregates customer demand per period. Lot sizes are generated on an aggregate level, and by applying a filling algorithm the sizes for batches on item level are derived. Setups are implicitly modeled by reducing available capacity. For problems with 15 product families and 10 time periods, the best heuristic of Özdamar and Bozyel (2000) deviated only 1.6% from the optimal solution with an average CPU time of 5 seconds per problem.

Problem-specific greedy heuristics

Greedy heuristics are myopic solution procedures. They generate solutions by working through consecutive periods. The lot sizes are determined in a step-by-step manner. These steps comprise the adaption of the lot sizes by increasing them based on priority criteria for different items. The usual suspects for these criteria are heuristics such as the Silver-Meal heuristic, least unit cost, and part-period balancing. A feasibility check is incorporated to guarantee solutions that satisfy all capacity constraints and rule out backordering.

Constructive greedy heuristics generate solutions from scratch. The lot sizes are determined by only considering the current and prior/subsequent period (Eisenhut, 1975). In the context of rolling horizon scheduling, this shortsighted planning method can prevent nervousness. However, solutions can be far from optimal and there are no quality guarantees.

Alternatives are the improvement heuristics. They improve initial (infeasible) solutions by simple operations. These operations mainly concern shifting production quantities between different items and/or periods. An application to a multi-level CLSP is given by Boctor and Poulin (2005). However, they restrict the lot sizes to be of equal size in each level of the supply chain. In each iteration of the heuristics, the leftover capacity is computed for each level. The lot size is increased when this leftover is strictly positive. If there exists a deficit on capacity, the lot sizes in previous periods are increased. For a final improvement step, the heuristic merges entire lot sizes by shifting them to previous periods. The paper by Clark and Armentano (1995) provide an application of improvement heuristics for solving the MLCLSP. They use a similar kind of shifting procedure as Boctor and Poulin (2005), but their method does incorporate the product structure and does not require equally sized batches. Priority criteria for sorting items are based on capacity violation penalties. Based on this ordering, the forward and backward production shifts are applied to obtain an adequate planning. Overall, the best heuristic of Boctor and Poulin (2005) had an average deviation of 3.43% from the optimal solution and a maximum optimality gap of 8.40%. However, these type of heuristics are significantly faster than their mathematical programming counterparts.

Metaheuristics

This class of heuristics combine the detailed knowledge of problem-specific heuristics, as discussed in the previous paragraphs, with a coordinating search routine. I.e., these heuristics use local knowledge to generate small moves in the search space and a standardized routine to select and regulate these movements. Compared to regular greedy heuristics, metaheuristics explore the search space more extensively, i.e., they avoid getting trapped in local optima.

Variable neighborhood search consists of three actions (Hansen & Mladenović, 2001). First, a random neighborhood of solutions is selected with respect to the current solution. Consecutively, a local search is applied on this neighborhood. The best move is compared to the current solution. If the former outperforms the latter, then the move is applied. Simulated annealing is a different probabilistic algorithm (Kirkpatrick, 1984). It decides to move to a neighbouring solution with a specific probability. These probabilities depend on the difference in solution quality and a time-varying parameter. The latter parameter is named the temperature. The temperature parameter controls the convergence of the algorithm to a local, and maybe even global, optimum. Helber (1995) discusses a variety of constructions regarding the temperature parameter for optimizing the MLCLSP. He also studied the effect of generating multiple starting solution on the quality of the final solution. Another method is Tabu search (Glover, 1986). The procedure is founded upon a list construction that collects data on moves that were executed. These moves cannot be reversed. The size of the list determines the search space.

Another class of heuristics to consider are the evolutionary algorithms. Genetic algorithms

mimic natural selection. Solutions are encoded as chromosomes. The performance of a specific solution is called its fitness. The solutions with highest fitness remain in the population, and new generations of solutions are created by applying stochastic operations on their fittest predecessors. These operations select suitable chromosomes, combine them, or adjust them slightly by mutation. This algorithm is commonly combined with other algorithms. Özdamar and Barbarosoğlu (1999) combine this method with simulated annealing. In their case, simulated annealing is used to prevent the natural selection operations from getting stuck in a specific neighborhood. A similar group of evolutionary algorithms is the collection of memetic algorithms. They benefit from the incorporation of specific problem characteristics.
Finally, ant colony optimization encompasses a technique that resembles the actions of ants in real life. It models the pheromone trails that ants excrete when collecting food. Pitakaso, Almeder, Doerner, and Hartl (2005) implemented an ant colony algorithm for solving the ML-CLSP. They combined the algorithm with a fix-and-relax heuristic. The ant colony algorithm determines the setup variables that are fixed, and the resulting MIP is then solved. The pheromone information represents which subsets of binary variables are compelling to fix. The method of Pitakaso et al. (2005) improves significantly on the performance of the algorithms of Stadtler (2003) and Tempelmeier and Derstroff (1996), while using computation time limits corresponding to that of the former.

2.4 MLCLSP under Uncertainty

This section discusses the stochastic variant of the MLCLSP, and it introduces an assortment of different solution methods for the S-MLCLSP.

2.4.1 Solution Methodologies

This section is structured into three parts. First, we discuss the lack of stochastic, analytical models that can cope with the increased complexity by adding capacity restrictions and lot-sizing decisions. Secondly, stochastic programming solutions are discussed. Lastly, some theory about robust optimization is treated, and we consider its applicability to the stochastic MLCLSP.

Stochastic analytical models

One approach to solving our problem is the use of multi-echelon inventory models. In our problem, three aspects are included that significantly increase the complexity of these types of models. Finite capacity, lost sales and lot sizing induce intractability of stochastic production-inventory models (De Kok et al., 2018).
A majority of the literature still assumes infinite capacity (De Kok et al., 2018). Infinitesimal perturbation analysis avoids the complexity of the resulting analytical model (Glasserman & Tayur, 1995). At first, it generates sample paths with the specific inventory settings, and subsequently computes a stochastic gradient. This gradient is associated to a specific supply chain cost function. By numerical optimization procedures utilizing this gradient, the optimal base-stock levels are computed. In the work of Janakiraman and Muckstadt (2009), a new inventory policy is introduced that enables the analysis and optimization of serial multi-echelon inventory systems. They propose the multitier base-stock policy which is optimal for serial systems with the same capacity at all locations, deterministic lead times, and one unit of demand for each customer. Note that this research is limited to serial systems and restrained by several assumptions. Hence, the method seems impractical for real-life cases. The additional complexity seems to arise from losing the Markov property of the multi-echelon system (De Kok et al., 2018). Echelon inventory positions and stocks of different items are related over time. When there is infinite capacity, one can relate these variables for items and their predecessors at a fixed point in time. However, finite capacity extends this dependency over multiple time periods, and thereby the analysis gets considerably more complicated.
In the case of stochastic demand, lot sizing is also a factor that affects the tractability of

exact stochastic models. Approximately 80% of multi-echelon inventory research neglects this concept (De Kok et al., 2018). Lot sizing introduces a renewal aspect to the timing of orders. Rosling (1989) has shown how to translate serial systems into convergent assembly systems. This theory can be used to include lot sizing by incorporating nesting. In their work, Karaarslan, Kiesmüller, and De Kok (2013) dispute the generalization of this extension to general supply chains. The nesting concept is contradicted by the fact that important components are ordered in smaller lot sizes than non-key items while the latter ones have a shorter lead time. The orders of the former are smaller due to the parts being more expensive, and therefore ordered more frequently.
Another complication arises when unfulfilled demand gets lost. When assuming lost sales instead of backorders, we lose nice regenerative properties of the inventory process. In the case of backordering, the echelon inventory positions provide us with all necessary information to describe the underlying state space. However, when lost sales are assumed, we also need to keep track of all orders that are placed each period up till one lead time in the past. Hence, the state space of the underlying Markov decision process becomes more complex, and therefore nicely structured optimal policies probably do not exist (De Kok et al., 2018).

Multi-stage stochastic programming solved by SAA
There exist a variety of different stochastic programming approaches to deal with problems similar to the S-MLCLSP. In this section, we comment on the most used and comprehensive class of methods, i.e., stochastic programming models solved with sample average approximation. Note that the MP formulation is used throughout this section. For a comprehensive discussion of dynamic programming formulations and their time complexity, see Minoux (2018).
As a benchmark, we used the model of Brandimarte (2006), which adapted their deterministic model into a MIP with a scenario tree. Another possibility is generating a couple of series of realizations for the entire planning horizon (Hu & Hu, 2018).
Their exist several methods to tackle these types of models. One of the first efficient solution methodologies was provided by Balasubramanian and Grossmann (2004). They created an adaptive approach which translates a two-stage stochastic program into a rolling-horizon scheme. Beraldi, Ghiani, Grieco, and Guerriero (2006) used a fix-and-relax heuristic, as discussed in Subsection 2.3.1, by implementing it in an exact multi-stage stochastic program. Their model accounted for uncertainty in processing time and sequence dependent setups. In later work, Beraldi, Ghiani, Grieco, and Guerriero (2008) incorporated additional constructs into their model such that industrial applications could be solved. These constructs included rolling-horizon scheduling and parallel machines.
The underlying solution methodology is the SAA approach (Kleywegt, Shapiro, & Homem-de Mello, 2002). This procedure solves an arbitrary number of stochastic problems which are related by a number of scenarios. The stochastic problems are solved to optimality, and thereafter one can determine bounds for the true objective functions by reevaluation of the found solutions under a new set of scenarios. The expected cost function of the original set of problems provides the lower bound, and the reevaluated solution with the lowest cost provides the upper bound. In their work, Taş, Gendreau, Jabali, and Jans (2018) solve the deterministic CLSP with several simple heuristics and apply the SAA procedure. Their approach delivers high quality solutions with limited computation times.
Later on, Hu and Hu (2018) created a scenario generation and reduction technique, and they combined this method with a stability test to test the fit of the scenario sample. Fast forward selection is applied for reducing the sample size and generating an appropriate fit to the underlying probability measure. The paper solves the lot sizing and scheduling problem formulated as a multi-stage stochastic programming problem. The system is subject to demand uncertainty. The model does not reflect the dependency between demand and time periods, i.e., does not reflect conditional relationships. This is a substantial weakness of this approach. Curcio et al. (2018) avoid this weakness by applying building a scenario tree where the different scenario nodes are interconnected. They apply approximate and adaptive procedures to the GLSP with uncertainty in demand by modeling this problem with a multi-stage stochastic

programming model. It turns out that the adaptive strategies significantly improve the performance of stochastic programming solution methods. The approximate methods provide only slightly worse results but save a considerable amount of computation time.

Robust optimization
Finally, we discuss an alternative for the stochastic programming approaches. Stochastic programming is often characterized as an intractable approach for large MIP problems (Alem et al., 2018). Therefore, it prohibits efficiently dealing with combinatorial problems, and this especially holds for problems with more complex stochastic behavior. Under this circumstance, the set of scenarios is usually large and consists of several complicating dependencies. Furthermore, it might be difficult to generate a suitable scenario set from historical data, or the assumptions underlying the generation techniques might be too restrictive for practical cases. We will introduce robust optimization to counter the mentioned disadvantages of the scenario generation process when applying the SAA method. These approaches provide general solutions which are insensitive to variation, i.e., approximately optimal solutions with the property to remain feasible after small adjustments in parameter values. For this purpose, these type of methods capture stochastic parameters in uncertainty sets (Ben-Tal, Ghaoui, & Nemirovski, 2009). These sets can be defined by utilizing only limited information on the underlying probability space. Robust optimization models are of a general form given by

$$\min_{\mathbf{x}} \{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{d}\}_{(\mathbf{c},\mathbf{A},\mathbf{d})\in\mathcal{U}}, \tag{2.7}$$

where the input coefficients are all uncertain, reals, and part of the uncertainty set $\mathcal{U}$. This boils down to min-max optimization over the worst combinations of parameter settings. The essence of this approach lies in the specification of an uncertainty set. Bertsimas and Sim (2004) propose a polyhedral uncertainty set

$$\mathcal{U} = \left\{ \mathbf{D} \in \mathbb{R}_+^{K\times T} \mid \xi_{kt}^d \in [-1,1], \sum_{\tau=1}^{t} |\xi_{k\tau}^d| \leq \Gamma_{kt}, \forall k \in \mathcal{K}, t \in \mathcal{T} \right\}, \tag{2.8}$$

which is especially useful for our setting, as it limits the intrinsic conservatism of robust optimization methods. Note that this uncertainty set is associated to a random vector representing customer demand. This can trivially be generalized to the other stochastic parameters in our model. The $\xi_{j\tau}^d$ is a scaled demand deviation, which is based on the maximum deviation in the realization interval. The Γ_{kt} parameter controls the values of coefficients. It induces that only a limited amount of parameters can take on their boundary values and places restrictions on the maximum deviations. This parameter is also called "the budget of uncertainty" and it mirrors the risk adversity of the decision maker.
Another important concept in solving the robust optimization model is the robust counterpart (RC), which is given by

$$\min_{\mathbf{x}} \{\mathbf{c}^T\mathbf{x} : \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \leq \mathbf{d}\ \forall \boldsymbol{\xi} \in \mathcal{Z}\}. \tag{2.9}$$

$\mathcal{Z}$ is a primitive uncertainty set derived from the original set $\mathcal{U}$, which describes specific parameter settings. A solution $\mathbf{x}$ is feasible when the constraint-wise uncertainty $\{\mathbf{A}(\boldsymbol{\xi})\mathbf{x} \leq \mathbf{d}\}$ is satisfied $\forall \boldsymbol{\xi} \in \mathcal{Z}$. It is possible to rewrite the original problem to its RC by incorporating the aforementioned uncertainty sets in the MIP model (Melamed, Ben-Tal, & Golany, 2016; Alem et al., 2018). After rewriting the original problem, the resulting MP problem can be solved.
Different variants of robust optimization are frequently applied for a selection of different problems, see Gabrel, Murat, and Thiele (2014) for an extensive review on different robust optimization application areas. Gorissen, Yanıkoglu, and Den Hertog (2015) mention a couple of factors that affect the quality of a robust solution. First and foremost, the chosen uncertainty set is a crucial factor for the performance of the optimization. The set has to resemble the real-life context. Furthermore, the probability distribution that underlies the stochastic process can also influence the performance of this methodology. If an inferior set is chosen

to protect against the parameter uncertainty, this could deteriorate the aforementioned resemblance to the real-life situation. Hence, when an uncertainty set assumes a symmetric geometric shape, but the underlying distribution is highly skewed, this could drastically affect the robust solution's quality. Furthermore, note that even though two different formulations of a mathematical problem are equivalent their robust counterparts do not necessarily have to be too, and therefore these models could return different solutions.
Ben-Tal, Goryashko, Guslitzer, and Nemirovski (2004) introduce a useful optimization paradigm called adjustable robust optimization (ARO). In these type of models, a specific set of variables can be adjusted after realization of stochastic parameters. This approach turns out to be less conservative than regular robust optimization. Unfortunately, problem instances generated by ARO are usually NP-hard. This problem is usually fixed by only considering solution values that lie in the affine space of the uncertain data. This typically results in more manageable problems, i.e., LP problems or semidefinite programming problems. Ben-Tal et al. (2004) apply their methodology on a production-inventory control problem.
ARO was used to solve a single new product multi-period planning problem by Melamed et al. (2016). The demand is assumed to follow a lifecycle pattern and its stochastic behaviour is captured by an uncertainty set. The target is to optimize expected profit over the product lifecycle. Curcio et al. (2018) extend the literature on ARO applications in lot sizing and production scheduling. They apply ARO on the GLSP with demand uncertainty. In addition, they apply approximate techniques by introducing affine functions of the uncertainty parameters. The approximate methods turn out to return good quality solutions with acceptable computation times even for large instance sizes.
Finally, another robust optimization method worth considering is distributionally robust optimization. Postek, Ben-Tal, Den Hertog, and Melenberg (2018) derive production-inventory models by applying this optimization methodology. Instead of using the variance as a dispersion measure, they opt for the mean absolute deviation from the mean (MAD). Using mean and dispersion information, several bounds were derived for the worst-case expectation. The DRO paradigm revolves around the so-called ambiguity set $\mathcal{P}$. The random parameter vector $\mathbf{z}$, which constitute the uncertainty set in robust optimization, is now captured by a probability measure $\mathbb{P}_{\mathbf{z}} \in \mathcal{P}$. For DRO optimization, we need to minimize

$$\sup_{\mathbb{P}_{\mathbf{z}} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}_{\mathbf{z}}}[f(\mathbf{x}, \mathbf{z})], \tag{2.10}$$

which boils down to optimizing the worst-case expectation. The set $\mathcal{P}$ consists of all distributions which match the mentioned mean and dispersion parameters and adhere to a predefined support. Hence, $\mathcal{P} = \{\mathbb{P} : supp(z_i) \subseteq [a_i, b_i], \mathbb{E}[z_i] = \mu_i, \mathbb{E}|z_i - \mu_i| = d_i, \forall i, z_i \perp z_j, \forall i \neq j\}$, where μ_i is the expectation of the uncertain parameter z_i, d_i is its MAD, and a_i and b_i represent the lower and upper bound of the support, respectively. We assume that the uncertain parameters are stochastically independent. In the work of Postek et al. (2018), expressions are derived to bound the expression given by (2.10). This new construct is considered quite useful, because it preserves the worst-case protection property of robust optimization without being over conservative.

2.5 Research Gaps

This section will provide a concise conclusion to our literature review by summarizing the main findings and pointing out missing content in the research of the S-MLCLSP.

Multi-echelon production-inventory planning subject to uncertainty in demand
As this literature study shows, the deterministic variant of the MLCLSP is already studied extensively. A plethora of papers describe different formulations, extensions, and solution methodologies (Billington et al., 1983), and in addition they describe the application areas regarding supply chain structures and their production-distribution planning (Erenguc et al., 1999). Furthermore, an extensive collection of literature exists on solving the single-level capacitated lot-sizing problems, which is extended to the multi-level situation. By applying

general optimization techniques, such as Lagrangian relaxation, problem reformulation, and decomposition, the MLCLSP can be reduced to more tractable, smaller problems. By applying Lagrangian relaxation on the capacity and goods flow contraints, one reduces the MLCLSP to uncapacitated single-level problems, which can be solved in polynomial time. Reformulation and decomposition also intend to simplify the problem and produce suitable lower bounds. Multiple papers describe the performance of these algorithms in terms of computational effort and optimally gaps. Additionally, metaheuristics are used to enhance these properties.
When considering the stochastic MLCLSP, it turns out that existing literature is considerably less complete. In essence, most of the discussed solution methodology boils down to the application of basic two-stage stochastic programming, LP-based methods incorporating safety stocks, simulation combined with metaheuristics for parameter optimization, elementary robust optimization techniques, or the derivation of analytical models subject to several simplifying assumptions, e.g., modeling the capacitated resources with naive queueing models. However, several authors already applied more sophisticated algorithms, such as ARO and multi-stage stochastic programming, to the GLSP (Curcio et al., 2018; Alem et al., 2018). Hence, as mentioned by Curcio et al. (2018), an interesting topic for future research is the application of the latter techniques in the multi-level environment.

Integrality issues
Another aspect of the problem which requires additional attention is the modeling of setup decisions with binary decision variables. The associated setup times induce the NP-hardness of the MLCLSP. This complexity is one of the main concerns discussed in the papers on deterministic MLCLSP, which were addressed in the previous paragraph. However, the introduction of uncertainty will exacerbate the impact of the setup concept. In general, MIP-based algorithms in the rolling-horizon context introduce here-and-now and wait-and-see variables when subject to uncertainty. Furthermore, several approaches introduce new sets of constraints and variables, e.g., the scenarios in SAA. Hence, the curse of dimensionality inflates the problem size and deteriorates the tractability of the MLCLSP.
Current literature resolves these issues by using a fix-and-relax heuristic (Beraldi et al., 2008; Curcio et al., 2018). The here-and-now decisions are restricted to binary decision variables. On the other hand, setup decisions in future periods are relaxed or even omitted. Research regarding the application of efficient relaxation and/or decomposition methods on the stochastic MLCLSP is absent. Future research could consider the structural properties that arise from the scenario generation inherent to multi-stage stochastic programming, or use relaxation techniques on the underlying deterministic MLCLSP. These approaches could reduce computation times and increase cost performance compared to the fix-and-relax heuristic.
In the case of ARO, wait-and-see variables are introduced to tackle uncertainty revealed in future stages of the problem. These wait-and-see variables induce intractability, which is resolved by restricting the wait-and-see variables to be affine functions of the uncertain demand (Melamed et al., 2016). However, this approach fails when the wait-and-see variables are restricted to be integers. Integer variables cannot be modeled by mapping parametric functions onto them. Yanıkoglu et al. (2018) mention this complication when solving an MIP-based ARO model. There are approaches to circumvent this issue by partitioning the uncertainty set. However, these methods introduce auxiliary integer variables, which unfortunately increase computational complexity even further. Potential research lies in the application of Lagrangian relaxation techniques to manage the increase in problem complexity.

Stochastic programming and robust optimization extensions
Different extensions of stochastic programming and robust optimization are of interest for future research. An important topic to consider when applying multi-stage stochastic programming is the method for generating scenarios. The papers of both Curcio et al. (2018) and Alem et al. (2018) use different techniques for generating scenarios. These different methods should be compared on their performance. Another topic of interest is the application of (adjustable) quadratic robust optimization. Spitter (2005) has shown that quadratic objective

functions for the SCOP model can induce superior solutions. Nonlinear adjustable robust optimization, however, is an extremely hard topic (Yanıkoglu et al., 2018). Furthermore, another nice extension of robust optimization is recoverable robust optimization. This technique is used in the paper of (Van den Akker, Bouman, Hoogeveen, & Tönissen, 2016). The main idea is the construction of an algorithm that restores original solutions when they become feasible again in later stages. Van den Akker et al. (2016) apply this new method to the knapsack problem.
Finally, the DRO concept discusses in this chapter might also proof useful when solving the S-MLCLSP. Future research should try to employ these methods to different types of higher-level planning problems.

Practical issues in industry
Tempelmeier (2001) observes the importance of LP-based models in current-day APS modules. The author remarks that one of the most disappointing characteristics of current APS is their inability to handle lot sizing issues. Implementing practical SCOP-based models is still an issue. Introducing uncertainty in model parameters only exacerbates these problems. Furthermore, De Kok and Fransoo (2003) remark that empirical validation of the SCOP models is still an open topic for research. For these models to have any practical value, the SCOP concepts should be validated in industry. In industry, a selection of different assumptions made in literature generally do not hold. E.g., the backordering assumption and ergodicity of the system. When optimizing transient behaviour of a supply chain system, one cannot rely on steady-state analysis. Hence, the initial state of the system becomes an essential part of the analysis. This renders techniques such as the one introduced in the paper of Kohler-Gudum and De Kok (2002) ineffective as the system is assumed to be ergodic. As such, the transient behavior is an important topic for future research.

Chapter 3

Derivation of Optimization Models

A small overview of the solution approaches is provided in this section. Note that we first derive models for the case without MOQ constraints. We discuss the following heuristic procedures: a deterministic LP-based rolling-horizon planning, an LP model with safety stocks, a robust optimization model, sample average approximation, and a distributionally robust optimization model. In particular, this section is focuses on ways to model the uncertainty in demand that emerges in the end-item stock points. Constraints regarding resource capacity restrictions and material coordination for intermediate items are almost identical between the different models.

3.1 Deterministic LP-based Rolling-Horizon Planning

This is the standard optimization model that served as the benchmark, which is based on the model stated in Subsection 1.3.1. The optimization model is given by

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} [h_k I_{kt} + b_k L_{kt}] \tag{1.1}$$

$$s.t. \quad I_{k,t-1} + Q_{k,t-z_k} - \sum_{j \in \mathcal{S}_k} a_{kj} Q_{jt} = I_{k,t} \qquad \forall k \in \mathcal{K} \backslash \mathcal{E},\ \forall t \in \mathcal{T} \tag{1.2}$$

$$I_{e,t-1} + L_{e,t} + Q_{e,t-z_e} = I_{e,t} + \mathbb{E}[d_{et}] \qquad \forall e \in \mathcal{E},\ \forall t \in \mathcal{T} \tag{1.3}$$

$$\sum_{k \in \mathcal{K}_m} p_k Q_{kt} \leq c_{mt} \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1.4}$$

$$Q_{kt}, I_{kt}, L_{kt} \geq 0 \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{1.8}$$

The model specifics are already discussed in Subsection 1.3.1. This approach simply substitutes the demand outcome d_{et} with the expected value of the demand $\mathbb{E}[d_{et}]$. Furthermore, we drop Constraints (1.5), (1.6), (1.7), and (1.9) as these are related to the MOQ variant of our problem.

3.2 Safety Stock Heuristic

As the model described above in no way models the uncertainty in demand, we also implement an LP-based model with safety stocks. The safety stocks are incorporated to take care of the demand variability. Hence, the original production-inventory planning optimization model is slightly adapted to establish a more suitable benchmark for testing the performance of our main optimization approaches, which will be discussed in the remainder of this section. The safety stock model is based on the work of Absi and Kedad-Sidhoum (2009) and is given by

$$\min \quad \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\left[s_k^+ S_{kt}^+ + s_k^- S_{kt}^- + b_k L_{kt}\right] \tag{3.1}$$

$$s.t. \quad I_{kt} = S_{kt} + S_{kt}^+ - S_{kt}^- \qquad \forall k\in\mathcal{K},\ \forall t\in\mathcal{T} \tag{3.2}$$

$$S_{k,t-1}^+ + S_{kt}^- + Q_{k,t-z_k} - \sum_{j\in\mathcal{S}_k} a_{kj}Q_{jt} = S_{kt}^+ + S_{k,t-1}^- \qquad \forall k\in\mathcal{K}\backslash\mathcal{E},\ \forall t\in\mathcal{T} \tag{3.3}$$

$$S_{k,t-1}^+ + S_{k,t}^- + L_{e,t} + Q_{e,t-z_e} = S_{kt}^+ + S_{k,t-1}^- + \mu_{et} \qquad \forall e\in\mathcal{E},\ \forall t\in\mathcal{T} \tag{3.4}$$

$$S_{kt}^- \leq S_{kt} \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{3.5}$$

$$\sum_{k\in\mathcal{K}_m} p_k Q_{kt} \leq c_{mt} \qquad \forall m\in\mathcal{M}, \forall t\in\mathcal{T} \tag{3.6}$$

$$Q_{kt}, S_{kt}^+, S_{kt}^-, L_{kt} \geq 0 \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{3.7}$$

The objective function (3.1) now includes penalty costs for discrepancies between the desired safety stock level S_{kt} and the actual inventory on hand. A surplus is denoted by the variable S_{kt}^+ and penalized with a cost s_k^+ per unit, and a deficit S_{kt}^- incurs s_k^- for each unit short. Constraints (3.2) express the relationship between the inventory on hand and the safety stock variables. Constraints (3.3) and (3.4) impose the balancing of the safety stock level. Finally, Constraints (3.5) ensures that the actual safety stock deficit does not exceed the desired level. Note that this reformulation preserves the feasibility of solutions for the original problem. s_k^- needs to have a lower value than p_e for all end items. Otherwise, the adapted inventory balance equations will move safety stock shortages to actual shortages.

3.3 Robust Optimization (RO)

In our research, we base our RO optimization concept on the work done by Ben-Tal, Ghaoui, and Nemirovski (2009), Ben-Tal, Boaz, and Shimrit (2009), and Bertsimas and Thiele (2006). The RO version of our S-MLCLSP tackles demand by introducing the uncertainty sets described in Subsection 1.3.1. Robust optimization aims at generating the solution that provides the best protection to the uncertainty in our planning problem. Hence, it acts from a worst-case minimization perspective. We derive a model similar to the one in (Alem et al., 2018). However, the latter model assumes demand backordering. Therefore, a new model has to be constructed that is suitable for lost-sales systems.

First of all, we provide a couple of equivalent formulations for the cumulative amount of lost sales.

Lemma 3.3.1. *The cumulative amount of lost sales in the S-MLCLSP model is given by*

$$L_t^{cum} = (((((d_t - Q_t)^+ + d_{t-1} - Q_{t-1})^+ ...)^+ + d_2 - Q_2)^+ + d_1 - Q_1)^+, \tag{3.8}$$

where $(x)^+ := \max\{x, 0\}$*, which can be computed by using the following recursive expression:*

$$\hat{L}_t^{cum} := \begin{cases} (d_t - Q_t) + \hat{L}_{t+1}^{cum}, & \textit{if} \quad d_t + \hat{L}_{t+1}^{cum} \geq Q_t \\ (d_T - Q_T), & \textit{if} \quad t = T, d_T \geq Q_T \\ 0, & o.w., \end{cases} \tag{3.9}$$

and recognizing that $L_t^{cum} = \hat{L}_1^{cum} - \hat{L}_{t+1}^{cum}$*, where* $\hat{L}_{T+1}^{cum} := 0$.

Proof. The first statement in the lemma follows from adding the shortages of each period and recognizing the fact that shortages in one period can be reduced by leftover inventory from earlier periods. The recursive formulation follows trivially from backward recursion. Here $\hat{L}_1^{cum}$ represents the cumulative amount of lost sales over the entire planning horizon T. Hence, $L_T^{cum} = \hat{L}_1^{cum}$ which we will use repeatedly from this point onward. ■

As such, we provide a recursive expression for the cumulative amount of lost sales that can be implemented in an LP model. In the remainder, we also need the following mathematical construct.

Definition 3.3.1 (Bi-affine function)**.** *A function* $f(x, y)$ *is called bi-affine, if* $f(x, y)$ *is affine for any fixed* x *and the function is affine for any fixed* y*.*

Now let us prove that the lost-sales expression is convex, which also implies convexity for our LP reformulation. This is due to the direct relation between Equations (3.8) and (3.9).

Lemma 3.3.2. *The cumulative amount of lost sales given by the function* $L_t^{cum} : \mathbb{R}_+^t \times \mathbb{R}_+^t \to \mathbb{R}_+$

$$L_t^{cum}(\mathbf{Q}, \mathbf{d}) = (((((d_t - Q_t)^+ + d_{t-1} - Q_{t-1})^+ ...)^+ + d_2 - Q_2)^+ + d_1 - Q_1)^+ \tag{3.10}$$

is convex in the decision variables $\mathbf{Q} = (Q_1, Q_2, ..., Q_t)$ *and uncertain parameters* $\mathbf{d} = (d_1, d_2, ..., d_t)$*, where* $(x)^+ = \max\{x, 0\}$*.*

Proof. This result follows from several other lemmas in convexity theory. In particular, one uses some of the preservation properties of multivariate functions. We will use the Convex Monotone Superposition Principle. Let $f(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x}))$ be a vector function on $\mathbb{R}^n$ with convex components $f_i : \mathbb{R} \to \mathbb{R}$, and let $F : \mathbb{R}^n \to \mathbb{R}$ be a monotonically increasing function, i.e., $z \leq z'$ implies that $F(z) \leq F(z')$. Then the superposition $F(f(\mathbf{x}))$ is convex on $\mathbb{R}^n$. Furthermore, we use that if $g, h : \mathbb{R}^n \to \mathbb{R}$ are convex functions, then so is $g + h$. And we use that the $(g)^+$ operator preserves convexity, when g is a convex function. (Boyd & Vandenberghe, 2004).
We first look at the two most inner $(x)^+$-functions. Define $f_1(Q_t, d_t) = (d_t - Q_t)^+$, $f_2(Q_{t-1}, d_{t-1}) = (d_{t-1} - Q_{t-1})$ and $F_{t-1}(x, y) = (x + y)^+$. Both the bi-affine functions $(d_t - Q_t)$ and $(d_{t-1} - Q_{t-1})$ and the max-function $(x)^+$ are convex. Furthermore, it is trivial to see that F_{t-1} is convex and monotonically increasing in (x, y). Hence, $F_{t-1}(f_1(Q_t, d_t), f_2(Q_{t-1}, d_{t-1}))$ is convex on $dom(f_1) \times dom(f_2)$. Now substituting $f_1 = F_{t-1}(f_1(Q_t, d_t), f_2(Q_{t-1}, d_{t-1}))$ and recursively applying the argument on F_{t-1},...,F_1 provides us with the result. ■

The same reasoning can be applied to the inventory on hand variable which provides us with the following corollary.

Corollary 3.3.2.1. *The amount of overage in period* t *given by the function* $I_t : \mathbb{R}_+^t \times \mathbb{R}_+^t \to \mathbb{R}_+$

$$I_t(\mathbf{Q}, \mathbf{d}) = (((((Q_1 - d_1)^+ + Q_2 - d_2)^+ ...)^+ + Q_{t-1} - d_{t-1})^+ + Q_t - d_t)^+ \tag{3.11}$$

is convex in the decision variables $\mathbf{Q} = (Q_1, Q_2, ..., Q_t)$ *and parameters* $\mathbf{d} = (d_1, d_2, ..., d_t)$*.*

Proof. This follows from similar arguments as the proof of Lemma 3.3.2. ■

Now we use these assertions to transform our original problem and include the uncertainty set to model the stochastic demand parameters. The uncertainty set is described in Subsection 1.3.1. Here μ_{et} represents the expected demand for end item e in period t. Furthermore, we assume $\hat{d}_{et}$ to be the maximum deviation of the demand parameter.

Lemma 3.3.3. *The material coordination problem at the final echelon in our original deterministic MLCLSP model can be rewritten in equivalent form to*

$$\begin{aligned}
\min_{\mathbf{Q}} \quad & \sum_{e\in\mathcal{E}}\sum_{t\in\mathcal{T}} h_e I_{et} + \sum_{e\in\mathcal{E}} p_e \hat{L}^{cum}_{e1} && && (3.12)\\
s.t. \quad & \hat{L}^{cum}_{et} \geq d_{et} - Q_{e,t-z_e} + \hat{L}^{cum}_{e,t+1} \geq 0, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.13)\\
& I_{et} \geq Q_{e,t-z_e} - d_{et} + I_{e,t-1} \geq 0, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.14)
\end{aligned}$$

which has the Robust Counterpart (RC)

$$\begin{aligned}
\min_{\mathbf{Q}} \quad & \max_{\boldsymbol{\xi}} \sum_{e\in\mathcal{E}}\sum_{t\in\mathcal{T}} h_e I_{et} + \sum_{e\in\mathcal{E}} p_e \hat{L}^{cum}_{e1} && && (3.15)\\
s.t. \quad & \hat{L}^{cum}_{et} \geq d_{et} - Q_{e,t-z_e} + \hat{L}^{cum}_{e,t+1} \geq 0, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.13)\\
& I_{et} \geq Q_{e,t-z_e} - d_{et} + I_{e,t-1} \geq 0, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.14)\\
& d_{et} = \mu_{et} + \hat{d_{et}}\xi_{et}, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.16)\\
& 0 \leq \xi_{et} \leq 1 && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.17)\\
& \sum_{\tau=1}^{t} \xi_{e\tau} \leq \Gamma_t && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E} && (3.18)
\end{aligned}$$

where the demand is assumed to lie in the uncertainty set

$$\mathcal{U}_d = \left\{ \mathbf{D} \in \mathbb{R}^{E\times T}_{+} \mid \xi^d_{et} \in [-1,1], \sum_{\tau=1}^{t} |\xi^d_{e\tau}| \leq \Gamma^d_{et}, \forall e\in\mathcal{E}, t\in\mathcal{T} \right\} \tag{3.19}$$

Proof. The equivalence of the deterministic MLCLSP model and our new model follows from Lemmas 3.3.1 and 3.3.2 and Corollary 3.3.2.1. The new objective function (3.15) is convex, as it is the sum of convex functions. Furthermore, note that the inequalities for the recursive Constraints (3.13) and (3.14) will always be tight. The RC follows from inserting the aforementioned uncertainty set. ■

Finally, we show that we can construct an LP model from the non-linear model mentioned above.

Theorem 3.3.4. *The RC is equivalent to the following problem*

$$\begin{aligned}
\min \quad & G && && (3.20)\\
s.t. \quad & \hat{L}^{cum}_{et\xi} \geq d^{\xi}_{et} - Q_{e,t-z_e} + \hat{L}^{cum}_{e,t+1,\xi}, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E},\ \forall \xi\in\Xi && (3.21)\\
& I_{et\xi} \geq Q_{e,t-z_e} - d^{\xi}_{et} + I_{e,t-1,\xi}, && \forall t\in\mathcal{T},\ \forall e\in\mathcal{E},\ \forall \xi\in\Xi && (3.22)\\
& G \geq \sum_{e\in\mathcal{E}}\sum_{t\in\mathcal{T}} h_e I_{et\xi} + \sum_{e\in\mathcal{E}} p_e \hat{L}^{cum}_{e1\xi}, && \forall \xi\in\Xi && (3.23)
\end{aligned}$$

where the ξ represent vertices of the uncertainty set.

Proof. Our reasoning follows the same lines as that from the paper of Gorissen and Den Hertog (2013). As the l.h.s. (analysis) variables I_{et} and $\hat{L}^{cum}_{et}$ are convex in the uncertain demand parameters, and a convex function always takes its maximum value at the extreme points of its domain, we conclude that the problem above gives the exact RC and is equivalent to the problem stated in Theorem 3.3.3. ■

The only thing left to do is adding the non end-item inventory balance constraints and the capacity restriction. Note that the extension only applies to the end items $e \in \mathcal{E}$, which are subject to stochastic demand. Note that we have derived the exact robust counterpart conform to the definition of Gorissen and Den Hertog (2013), which mention the fact that most papers often study conservative reformulations. As we have rewritten our original model with equivalent statements and we consider all vertices of the uncertainty set, it can be stated that our RC is an exact reformulation. The number of constraints in our model grows exponentially in the planning horizon T. Bienstock and Özbay (2008) show that exact formulations which incorporate the vertices of the uncertainty set are efficiently solvable. We use the Double Description method discussed in the research of Fukuda and Prodon (2012) to find the vertices of the uncertainty set. This method has polynomial running time in the input, which is the linear system that describes the uncertainty set polyhedron.

3.4 Sample Average Approximation (SAA): Multi-Stage Stochastic Programming Approximation

Robust optimization focuses on the worst-case scenario and tries to mitigate its risks. However, this might deteriorate the average performance. Therefore, this research also studies stochastic programming methodology for resolving our production-inventory planning problem (Shapiro, 1993). First of all, note that the exact stochastic programming model is given by

$$\min_{\mathbf{Q}_t \geq \mathbf{0}} \sum_{t \in \mathcal{T}} \sum_{t \in \mathcal{K}} \mathbb{E}_{\mathbb{P}} \left[h_k I_{kt} + b_k L_{kt} \right] \tag{3.24}$$

$$s.t. \quad (1.2),\ (1.4),\ \text{and}\ (1.5)\ \text{hold} \tag{3.25}$$

$$I_{e,t-1} + L_{e,t} + Q_{e,t-z_e} = I_{e,t} + D_{et} \qquad \forall e \in \mathcal{E},\ \forall t \in \mathcal{T} \tag{3.26}$$

where we take the probability measure $\mathbb{P}$ with respect to the stochastic processes $(D_e)_{t \geq 0}$, $\forall e \in \mathcal{E}$, which we model with the probability space $(\Omega_d, \mathcal{F}_d, \mathbb{P}_d)$. However, it should be noted that this formulation lacks a precise meaning. One way to realize this model is by introducing stochastic dynamic programming equations, such as

$$V_t(\mathbf{I}_{t-1}, \mathbf{d}_{t-1}) = \min_{\mathbf{Q}_{t-z} \geq \mathbf{0}} \sum_{k \in \mathcal{K} \setminus \mathcal{E}} h_k I_{kt} + \mathbb{E}_{\mathbb{P}}[\sum_{e \in \mathcal{E}} (h_e (I_{e,t-1} + Q_{e,t-z_e} - D_{et})^+ \tag{3.27}$$

$$+ b_e (D_{et} - I_{e,t-1} + Q_{e,t-z_e})^+) + V_{t+1}(\mathbf{I}_t, \mathbf{d}_t) | \mathbf{D}_{t-1} = \mathbf{d}_{t-1}] \tag{3.28}$$

$$s.t. \quad I_{k,t-1} + Q_{k,t-z_k} - \sum_{j \in \mathcal{S}_k} a_{kj} Q_{jt} = I_{k,t} \qquad \forall k \in \mathcal{K} \setminus \mathcal{E} \tag{3.29}$$

$$\sum_{k \in \mathcal{K}_m} p_k Q_{kt} \leq c_{mt} \qquad \forall m \in \mathcal{M} \tag{3.30}$$

However, note that this optimization model suffers from the curse of dimensionality. The state and action space grow exponentially with the planning horizon and number of items. Furthermore, demand should be discretized and bounded in the case of continuous demand to make the problem solvable by stochastic dynamic programming. To solve the stochastic programming model, we opt for an approximate procedure: the sample average approximation (Kleywegt et al., 2002).

Along the lines of Brandimarte (2006), we derived a scenario-based multi-stage stochastic programming model:

$$\min \quad \sum_{s\in\mathcal{S}} \pi_s \cdot \Big(\sum_{k\in\mathcal{K}} \big[h_k I_{k,s(t),s} + b_k L_{k,s(t),s}\big] \Big) \tag{3.31}$$

$$s.t. \quad I_{k,s(t-1),s} + Q_{k,s(t-z_k),s} - \sum_{j\in\mathcal{S}_k} a_{kj} Q_{j,s(t),s} = I_{k,s(t),s} \qquad \forall k \in \mathcal{K}\backslash\mathcal{E},\ \forall s \in \mathcal{S} \tag{3.32}$$

$$I_{e,s(t-1),s} + L_{e,s(t),s} + Q_{e,s(t-z_e),s} = I_{e,s(t),s} + d_{e,s(t),s} \qquad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S} \tag{3.33}$$

$$\sum_{k\in\mathcal{K}_m} p_k Q_{k,s(t),s} \le c_{mt} \qquad \forall m \in \mathcal{M}, \forall s \in \mathcal{S} \tag{3.34}$$

$$Q_{k,s(t),s}, I_{k,s(t),s} L_{k,s(t),s} \ge 0 \qquad \forall k \in \mathcal{K}, \forall s \in \mathcal{S} \tag{3.35}$$

The objective function (3.31) includes a weight factor which assigns probability π_s to a specific node $s \in \mathcal{S}$. One path in through the scenario tree represents a sample path realization. The inventory balance equations (3.32) and (3.33) now also include dependencies between different nodes based on the generated scenario tree. $s(t-i)$, $i = 1, ..., t$ represents the ancestor vertex of s that occurred i periods ago. Hence, the model links different nodes based on the underlying scenario tree. The capacity constraints (3.34) are also included for the different scenarios.
The scenario tree could be generated analogously to the work of Brandimarte (2006). The nodes of the tree are represented by the set $\mathcal{S}$. To create dependency between different scenarios/time periods, each scenario node s has a set of successor nodes $A(s)$. The number of successor nodes in period t is equal to $a(t)$. Hence, the total size of the scenario tree constituting of T periods is $\prod_{t=1}^{T} a(t) + 1$.

3.5 Distributionally Robust Optimization (DRO)

Distributionally robust optimization is a relatively new topic in the broader spectrum of RO (Gabrel et al., 2014). This methodology optimizes the worst-case expected performance over a (possibly infinite) set containing probability distributions. Contrary to RO, DRO protects against the worst-case probability measure instead of the worst-case parameter realization (Shang & You, 2018). Stochastic programming optimizes the expected value of the problem. However, this method requires knowledge of the underlying probability distribution, which is hard to estimate accurately by analyzing empirical data. Robust optimization only needs information about the values the uncertain parameters can attain, but this method could be too conservative and protect against highly improbable parameter values. Distributionally robust optimization balances these approaches and hedges against distribution fitting errors by taking the worst-case probability distribution.
For deriving our DRO optimization model, we apply mean absolute deviation from the mean bounds derived in previous research and the lemmas derived in Section 3.3. We use Proposition 1 from the work of Postek et al. (2018) and formulas (13) of their work.

Proposition 3.5.1. *If $f(\mathbf{x}, \cdot)$ is convex, it holds that*

$$\sup_{\mathbb{P}\in\mathcal{P}_{(\mu,\bar{d})}} \mathbb{E}_{\mathbb{P}}[f(\mathbf{x}, \mathbf{z})] = \sum_{\alpha\in\{1,2,3\}^T} \prod_{i=1}^{T} P^i_{\alpha_i} f(\mathbf{x}, \tau^1_{\alpha_1}, ..., \tau^T_{\alpha_T}), \tag{3.36}$$

where $P^i_{\alpha_i}$ and $\tau^i_{\alpha_i}$ are defined as

$$P_1^i = \frac{\tilde{d}_i}{2(\mu_i - l_i)},\ P_2^i = 1 - \frac{\tilde{d}_i}{2(\mu_i - l_i)} - \frac{\tilde{d}_i}{2(u_i - \mu_i)},\ P_3^i = \frac{\tilde{d}_i}{2(u_i - \mu_i)}, \tag{3.37}$$

$$\tau_1^i = l_i,\ \tau_2^i = \mu_i,\ \tau_3^i = u_i, \quad \text{for } i = 1, ..., T \tag{3.38}$$

Proof. The proof is given in the work of Postek et al. (2018). ■

Let us explain the expressions and new variables introduced in this proposition. We introduce an ambiguity set $\mathcal{P}_{(\mu,\tilde{d})}$, which contains the probability distributions with the corresponding partial distribution information: an expected value of μ and MAD $\tilde{d}$. We can bound the worst-case expectation with the expression given in (3.36). It can be shown that the worst-case distribution of the ambiguity set is given by a three-point probability distribution with the probability mass and support given by (3.37) and (3.38), respectively. Here l_i is the lower bound and u_i the upper bound of the parameter's support. For each of the T constraints we enumerate all possible values. Hence, we compute 3^T different permutation terms for the summation.
In the remainder, we combine the results of this proposition with the convexity of the cumulative lost-sales recursion and the inventory on hand expression, where the latter is proven in Lemma 3.3.2 and Corollary 3.3.2.1. This results in the following model/theorem:

Theorem 3.5.2. *The LP formulation of the DRO material coordination problem at the end-item stock points is given by*

$$\min \quad \sum_{e\in\mathcal{E}} \sum_{\alpha\in\{1,2,3\}^T} \prod_{i=1}^{T} P^i_{\alpha_i} \left(\sum_{t\in\mathcal{T}} h_e I_{et\alpha} + p_e \hat{L}^{cum}_{e1\alpha} \right) \tag{3.39}$$

$$s.t. \quad \hat{L}^{cum}_{et\alpha} \geq \tau^t_{e\alpha} - Q_{e,t-z_e} + \hat{L}^{cum}_{e,t+1,\alpha}, \qquad \forall t\in\mathcal{T},\ \forall e\in\mathcal{E},\ \forall\alpha\in\{1,2,3\}^T \tag{3.40}$$

$$I_{et\alpha} \geq Q_{e,t-z_e} - \tau^t_{e\alpha} + I_{e,t-1,\alpha}, \qquad \forall t\in\mathcal{T},\ \forall e\in\mathcal{E},\ \forall\alpha\in\{1,2,3\}^T. \tag{3.41}$$

Proof. Note that we use similar recursive expressions as for the RO model. Due to Lemma 3.3.2 and Corollary 3.3.2.1, we know that the expression $\sum_{t\in\mathcal{T}} h_e I_{et\alpha} + p_e \hat{L}^{cum}_{e1\alpha}$ is convex as it is the sum of convex functions. Furthermore, the problem is separable in the end items $e \in \mathcal{E}$. When applying DRO, we have to solve the following problem:

$$\sup_{\mathbb{P}\in\mathcal{P}_{(\mu,\tilde{d})}} \mathbb{E}_{\mathbb{P}} \left[\sum_{e\in\mathcal{E}} \sum_{t\in\mathcal{T}} h_e I_{et} + \sum_{e\in\mathcal{E}} p_e \hat{L}^{cum}_{e1} \right] = \sum_{e\in\mathcal{E}} \sup_{\mathbb{P}\in\mathcal{P}_{(\mu,\tilde{d})}} \mathbb{E}_{\mathbb{P}} \left[\sum_{t\in\mathcal{T}} h_e I_{et} + p_e \hat{L}^{cum}_{e1} \right]$$

Now taking $f(\mathbf{x}, \mathbf{z}) = \sum_{t\in\mathcal{T}} h_e I_{et} + p_e \hat{L}^{cum}_{e1}$ and applying Proposition 3.5.1 provides us with the result. ■

Finally, the intermediate item inventory balance constraints and the capacity restriction are added. This final LP model can be solved in a similar way as the previous optimization models. Note that the amount of constraints regarding the end item inventory balance is of the order $\mathcal{O}(3^T)$. This could induce memory allocation issues for larger values of T. However, the problem is still solvable in polynomial time.

Chapter 4

Methodology of Computational Experiments: LP-based SCOP

To address our research questions we executed a set of experiments. The methodology is provided in this section. Section 4.1 describes the simulation scheme which was used to evaluate the provided solutions. Section 4.2 provides a short introduction to different instances of the experiments. This experimental setup was designed for the model variant without MOQ constraints.

4.1 Monte Carlo Simulation

Experimental instances were executed for all proposed solution approaches. By using a Monte Carlo simulation, the performance of an optimization model was evaluated. A simulation run followed the rolling-horizon scheme. The simulation procedure is displayed in Figure 4.1. First of all, the end-item demand of the current time bucket was revealed based on the mean and coefficient of variation of that period's demand. The actual demand distribution was assumed to be gamma distributed. The expectation and variance were given for the r.v.s describing the demand of each period and item. Furthermore, these variables were assumed to be independent and stationary between time periods. Afterwards, the optimization model was solved and the generated solutions were implemented for this period. Next, the demand of period $t + 1$ was sampled, and the model was also optimized for this period. This procedure was continued until the final period of a planning horizon reached the end horizon H. Note that we used one of the uncertainty profiles discussed by Brandimarte (2006), which is depicted in Figure 4.2. This uncertainty profile models the fact that customer orders dominate at the beginning of the planning horizon. These orders provide known demand quantities. When time progresses the uncertainty increases and we need to rely on forecasts. This profile provides an extreme representation of this scheme. The rolling-horizon schedules were simulated a significant amount of times. We executed 1000 replications to generate statistically significant results in acceptable computation times. This threshold was determined by measuring the convergence of the empirical cumulative distribution functions of the total system costs. The seed was set to one specific value for all simulation experiments to generate the same random numbers when comparing the different systems/approaches. This resulted in positively correlated systems, and thereby this approach reduced inter-system variance.
For the methods that were discussed we varied the control parameters to obtain their full potential. For the LP with safety stocks, the desired safety stock level was determined by using the techniques developed by Diks and De Kok (1998) implemented in the ChainScope software. These safety stock levels should be optimal in divergent, uncapacitated, ergodic inventory systems with backordering. However, unfulfilled demand is lost. Hence, the safety stock levels returned by the ChainScope software were not optimal for the systems that we considered. Furthermore, we varied the safety stock penalty factors from 0.5 to 6 with a step size of 0.5.

This penalty factor determined the values of s_k^+ and s_k^- as a fraction of the holding cost h_k.

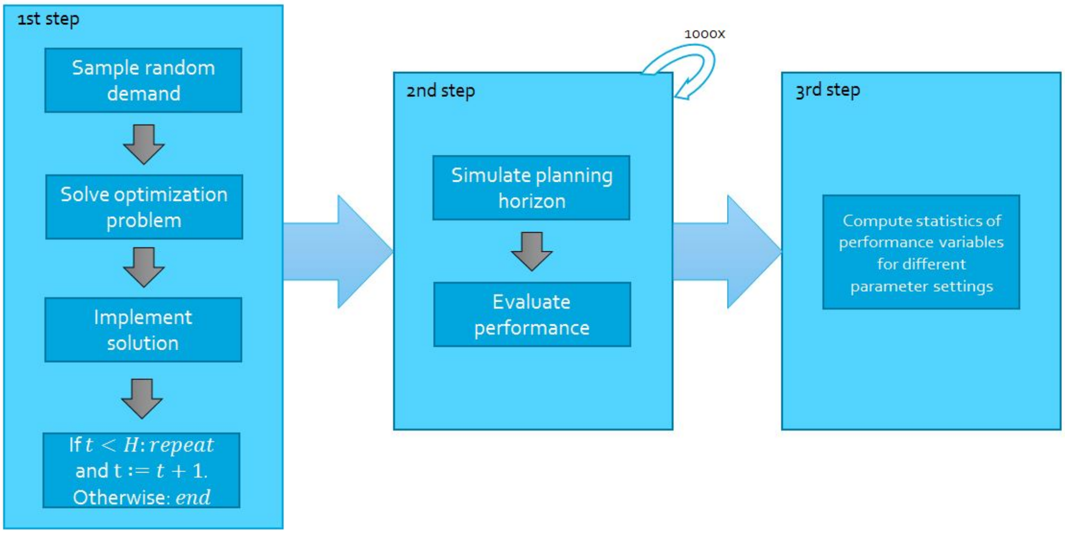


Figure 4.1: Monte Carlo simulation scheme with rolling-horizon scheduling

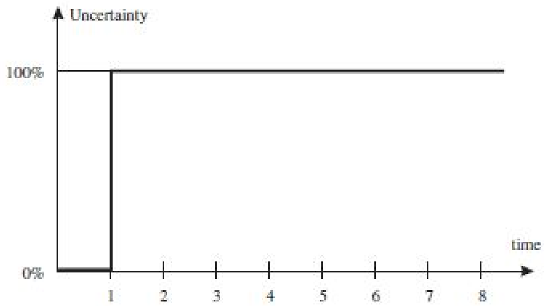


Figure 4.2: Uncertainty profile of demand (Brandimarte, 2006)

Similar to the work of Brandimarte (2006), the scenario generation strategy is rather elementary. The number of branches was determined by a branching factor, and after period t' this factor reduced to one. For each period an antithetic sampling procedure was applied, which is a well-know variance reduction technique. Symmetric samples were generated in the case of even numbered branching factors. In the case of an odd numbered factor, the samples were generated symmetric to the expected value, with one sample set to the expected value. For the size of the scenario tree, we chose to generate 691 nodes according to a specific structure. We started with one root node in period 1. Thereafter, we used a branching factor of 6 for the next 3 periods. From period 5 onward we applied a branching factor of 1. This specific setup returned 691 scenarios. This number of scenarios was chosen based on the overall solution quality and running time tractability. The scenario tree was structured according to the description in Section 3.4. We sampled from a truncated normal distribution equipped with the dispersion measures stated in Section 4.2. In Appendix A, the relationships between the moment parameters of the truncated normal and Gaussian normal distribution were derived. These relations were used in the scenario tree sampling procedure. Note that we assumed an alternative probability density for the sampling procedure with respect to the realized distribution. In practice only partial information about the demand distribution is known in advance, which normally only includes the first two moments, so we tried to mimic this in our experimental setup. We created a discrepancy between the assumed and realized distribution as in

practice distribution information is known only partially beforehand and thus actual demand data should elucidate the structure of the probability distribution. Hence, this simulation enabled us to include statements regarding the robustness/sensitivity of the stochastic approach regarding distributional information.
For the RO/DRO models, we had to define the upper and lower bounds for the support of the uncertain parameters. The experiments were executed with the variability level γ ranging in between $0.5-5$ and $0.5-3$, respectively, with steps of 0.25. Hence, the respective values were set to $\hat{d}_k = u_k - \mu_k = \mu_k - l_k = \gamma \cdot \sigma_k$, where σ_k is the standard deviation of item k's demand and we assure that $l_k \geq 0$. Note that for DRO it is also possible so set $l_k = 0$ and let $u_k \to \infty$. For the latter to be applicable, it needs to hold that $\lim_{u_k \to \infty} \frac{f(u_k)}{u_k} = L$ exists, where $f(u_k)$ is the value of the objective function at the upper bound (Postek et al., 2018). However, we used the same procedure to determine the support for both DRO and RO to make a more suitable comparison between these methods. Furthermore, we considered four types of "budget of uncertainty" functions Γ_{kt} similar to the ones in the research of Curcio et al. (2018): t, $0.5t + 0.5$, $0.1t + 0.5$, and $0.05t + 0.1$. The case $\Gamma_{kt} = t$ is the most conservative function. Here all the uncertain parameters attain their worst-case values. However, this approach might result in too conservative solutions. As such, more optimistic budgets were also implemented, which were already tested in previous research (Alem et al., 2018). E.g., if the budget is set to $\Gamma_{kt} = 0.5t + 0.5$, then the protection in the first period is equal to 1 and in the sixth period equal to 3.5. I.e., the first period is fully protected against the worst-case parameter realization, but the final period is hedged against approximately 58% of the coefficient variation. Finally, the MAD had to be calculated for the construction of the DRO model. Appendix B provides a formula for the MAD in the case of gamma distributed demand. For an extensive discussion on setting these tuning parameters, see the work of Alem et al. (2018).
We looked at different performance measures to compare the different optimization models: average cost, standard deviation of the cost and worst-case cost. The worst-case performance is defined as the maximum value found during a simulation run consisting of 1000 iterations. Furthermore, we kept track of the average lost sales, service levels, holding/penalty costs per period and capacity allocation. As such, this enabled the analysis of the system behaviour for the implemented optimization approaches. Moreover, we defined three metrics to compare the solutions in our simulation experiment. First of all, the Value of Modeling Uncertainty (VMU) was used to compute the relative gain of using a method that captures the uncertainty in the demand parameters compared to the deterministic LP model. It was computed by calculating the relative difference of the deterministic solution and the solutions of the other optimization methods. Let z^{avg}_{OPT} and z^{wc}_{OPT} be the average and worst-case performance, respectively, of model OPT. Then the VMU of a class relative to an optimization model is given by $VMU_{OPT} = \frac{z^{avg}_{LP} - z^{avg}_{OPT}}{z^{avg}_{LP}}$. Secondly, we constructed the Value of Stochastic Programming (VSP) metric, which was used to compare the average performance of the different methods with the SAA methods, as in theory this method should be optimal on average. This metric was calculated by taking the relative difference of the SAA solution with the other optimization approaches, i.e., $VSP_{OPT} = \frac{z^{avg}_{OPT} - z^{avg}_{SAA}}{z^{avg}_{SAA}}$. Finally, the third metric is the Value of Robust Optimization (VRO). This metric was used to compare the worst-case RO solution with the other worst-case solutions, as RO should be optimal in the worst-case. The following formula was used to compute this metric: $VRO_{OPT} = \frac{z^{wc}_{OPT} - z^{wc}_{RO}}{z^{wc}_{RO}}$.

4.2 Experimental Instances

To answer the first and second research question, we constructed an experimental test bed according to the structure of previous research (Curcio et al., 2018; Spitter, 2005; Alem et al., 2018). We considered the inventory-production system described by Figure 4.3 and constructed a factorial experimental design. The supply chain consisted of five different stock points ($K = 5$) and four different restricted resources ($M = 4$). PU2 was a shared resource, and PU3 and PU4 were chosen to be the bottleneck processes. This configuration was chosen for several reasons. First of all, this setup introduces a lag in decision responsiveness. Production at PU1 needs multiple time periods to fulfil demand that occurred at stock points 4 and 5. Furthermore, we introduce an allocation decision at PU2. At this resource a decision has to be made how to coordinate material that is positioned at stock point 1.

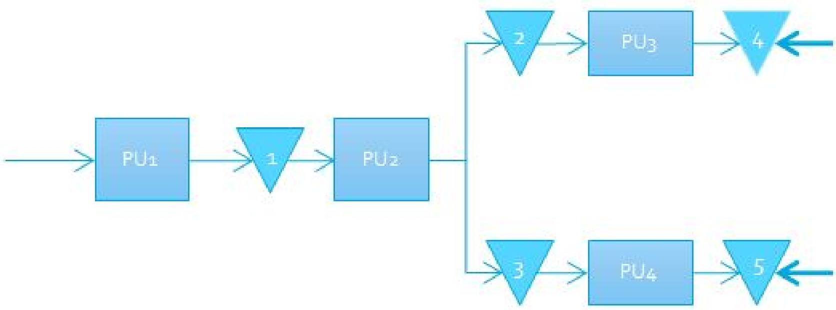


Figure 4.3: Supply chain under consideration

We organized the instances in several classes based on capacity, symmetry between end items, and coefficient of variation. First of all, we differentiated between different capacity levels on the bottleneck resources PU3 and PU4:

1. High capacity (low utilization): $c_t = \frac{\mathbb{E}[d_{et}]}{0.75}$
2. Medium capacity (medium utilization): $c_t = \frac{\mathbb{E}[d_{et}]}{0.85}$
3. Low capacity (high utilization): $c_t = \frac{\mathbb{E}[d_{et}]}{0.95}$

$\sum_{e \in \mathcal{E}} \mathbb{E}[d_{et}]$ is the sum of expected demand for all end items. Furthermore, we subdivided the instances in a group with high symmetry and a group with low symmetry between the end items. This distinction is based on the difference between expected demand and the holding/penalty costs of the final products. Furthermore, the impact of the coefficient of variation $cv \in \{0.25, 0.5, 1\}$ was studied. Table 4.1 displays all different classes in our setup.

Table 4.1: Different classes of the experiment and their attribute values

Classes	Utilization	Symmetry	cv
1	Low	Low	0.25
2	Low	Low	0.5
3	Low	Low	1
4	Low	High	0.25
5	Low	High	0.5
6	Low	High	1
7	Medium	Low	0.25
8	Medium	Low	0.5
9	Medium	Low	1
10	Medium	High	0.25
11	Medium	High	0.5
12	Medium	High	1
13	High	Low	0.25
14	High	Low	0.5
15	High	Low	1
16	High	High	0.25
17	High	High	0.5
18	High	High	1

For each of these classes the time horizon was set to $H = 12$ with a planning horizon $T = 6$. The other input parameters were controlled by fixing them in the following way:

1. Expected end-item demand (low/high symmetry): $\mathbb{E}[\mathbf{d}^{low}] = \begin{bmatrix} 300 \\ 100 \end{bmatrix}$, $\mathbb{E}[\mathbf{d}^{high}] = \begin{bmatrix} 200 \\ 200 \end{bmatrix}$
2. Holding costs (low/high symmetry): $\mathbf{h}^{low} = \begin{bmatrix} 0.5 \\ 1 \\ 1 \\ 2 \\ 5 \end{bmatrix}$, $\mathbf{h}^{high} = \begin{bmatrix} 0.5 \\ 1 \\ 1 \\ 2.75 \\ 2.75 \end{bmatrix}$
3. Penalty costs: $b_e = 10h_e$, $\forall e \in \mathcal{E}$
4. Processing times: $p_k = 1$, $\forall k \in \mathcal{K}$
5. Capacity non-bottlenecks PU1 and PU2: $c_{1,t} = c_{2,t} = 2\sum_{e\in\mathcal{E}} \mathbb{E}[d_{et}]$, $\forall t \in \mathcal{T}$
6. Planned lead times: $z_k = 1$, $\forall k \in \mathcal{K}$
7. Internal demands: $a_{kj} = 1$, $\forall j \in \mathcal{S}_k$, $\forall k \in \mathcal{K}$

For all these arrangements we measured the performance indicators of the different optimization methods by executing the aforementioned Monte Carlo simulation procedure.

Finally, we also studied the impact of initial inventory levels on the performance of the different optimization methods. In the cases described above, the initial inventory level I_0 was set equal to the pipeline inventory levels in the case of 100% fill rate. The initial inventory levels did not include safety stocks. This was done to initialize the system in a distorted state to study the way the system moves to a more stable situation. In additional experiments the level I_0 was varied as a fraction of this pipeline level. Moreover, the position of the bottleneck was shifted to PU1/PU2 in these additional experiments.

Chapter 5

Results of Computational Experiments

This chapter elaborates on the findings of our research. The experimental setup of our simulation study was given in Section 4.2. The structure of this chapter is as follows. First, we discuss the average and worst-case performance for all our methods. Secondly, we study the behaviour of our optimization approaches. Finally, results for the additional experiments are stated.

5.1 Average and Worst-Case Analysis

The results for all 18 different classes are stated in Table 5.2. An aggregate overview utilizing the measures we discussed in Section 4.1 is given in Table 5.1. As already mentioned, the (distributionally) robust optimization models were tested using two tuning parameters: the budget of uncertainty and the variability level. Hence, these two methods returned different performance distributions when shifting the focus between optimizing the average and the worst-case performance. Therefore, the empirical distribution function of the costs differed for (D)RO models initiated with distinct parameter settings. Originally, the safety stock model was also tuned by varying the parameter s_k^- and s_k^+, but this method proved insensitive to these changes. Note that we collected information about the underlying empirical distribution of the cost performance, as this measure is a random variable itself.

Table 5.1: VMU, VSP, and VRO for the different methods (all cases included)

	VMU	VSP	VRO	p-value DRO difference
LP	0.00%	41.57%	29.67%	< 0.001***
Safety stock	28.97%	0.46%	4.97%	< 0.001***
RO	29.40%	-0.14%	0.00%	< 0.001***
SAA	29.30%	0.00%	4.16%	< 0.001***
DRO	30.23%	-1.33%	1.40%	

*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Not surprisingly, Table 5.2 shows us that the benchmark LP model was outperformed by all alternatives that incorporated uncertainty in some way. This result holds for all classes of the instances. The solutions were on average approximately 30% more costly, were less stable regarding the standard deviation, and were outperformed on robustness by all other models as well. Note that introducing the safety stock levels, which are computed using the approach of Diks and De Kok (1998), already provided a significant improvement compared to the deterministic LP approach.

Table 5.2: Monte Carlo simulation results for all optimization methods

	LP			Safety stock			RO			SAA			DRO		
Class	Average	σ	Worst-case	Average	σ	Worst-case	Average	σ	Worst-case	Average	σ	Worst-case	Average	σ	Worst-case
# 1	8885	3111	21196	6007	2468	15394	5968	2287	15382	6059	2206	17723	5864	2277	16258
# 2	17469	6995	49638	12162	5533	38206	12227	5396	35432	11985	5019	35935	11871	5108	35383
# 3	33075	15014	94058	24151	12079	74575	24091	11122	73536	24220	11671	73408	23821	11405	73805
# 4	8886	3088	20844	6060	2484	17280	5998	2272	15606	6229	2326	16126	5867	2266	16289
# 5	17476	6941	50534	12148	5484	39062	12297	5262	36069	12371	5321	40181	11896	5110	36290
# 6	33094	14969	96388	24217	12089	79154	24255	11218	77404	24223	11689	77584	23915	11480	77404
# 7	8885	3111	21196	5973	2460	15667	5974	2195	14453	6052	2342	16812	5842	2308	14914
# 8	17469	6995	49638	12272	5684	38967	12157	5212	36447	12189	5423	35914	12011	5384	36424
# 9	33075	15014	94058	24371	12383	78715	24189	11956	78763	24213	12045	78763	24070	11913	78763
# 10	8886	3088	20844	6005	2462	16178	6042	2176	14922	6074	2339	16876	5862	2292	15333
# 11	17476	6941	50534	12326	5691	40423	12175	5215	36377	12176	5421	38828	12050	5368	37352
# 12	33094	14969	96388	24426	12357	82755	24279	11961	81777	24283	12003	81810	24150	11949	81777
# 13	8885	3111	21196	6185	2646	17698	6072	2553	17630	6079	2556	17662	6078	2582	17788
# 14	17469	6995	49638	12676	6048	41026	12504	5816	37771	12057	5607	36857	12489	5861	37672
# 15	33075	15014	94058	24787	12825	82831	24562	12465	82831	24534	12557	82831	24534	12505	82831
# 16	8886	3088	20844	6203	2651	17907	6108	2571	17880	6107	2565	17880	6096	2576	17880
# 17	17476	6941	50534	12645	5996	42062	12503	5788	36942	12500	5803	41005	12490	5819	38800
# 18	33094	14969	96388	24828	12802	85368	24588	12431	85423	24561	12549	85368	24562	12487	85423

It proved harder to compare the other optimization approaches as average and worst-case optimal performance depends on the specific class instance. Hence, we summarized the performance of these methods by using the measures we defined in the previous section: VMU, VSP, and VRO. Table 5.1 shows the average values over all cases for these metrics. The different cases were weighted equally when calculating the averages, as each case consisted of an equally sized sample. Furthermore, we tested whether the performance of a specific method was significantly different from the other alternatives. The p-values were calculated by using a Wilcoxon paired-sample signed-rank test, where significance indicates that the cost performance of one method is stochastically smaller than that of the other method.

Table 5.1 shows us that the SAA approach does not perform significantly better than the other models that incorporate uncertainty. Overall, it turned out that the SAA outperformed the RO and safety stock approaches, but the former was beaten by the DRO model. Theoretically, SAA should have outperformed all the other methods on average, as the stochastic programming paradigm is grounded upon minimization of expectation values. However, we were dealing with heuristic/approximate procedures here. First of all, there existed a discrepancy between the assumed distribution and theoretical distribution. Hence, the scenario generation procedure of SAA was biased. DRO is more robust to distribution fitting errors, as it only considers four specific parameters of the underlying distribution: the lower/upper bound of the support,the mean of the r.v., and its mean absolute deviation from the mean. Moreover, SAA is an approximate solution procedure for the original stochastic program. These could be reasons, i.a., that DRO was able to outperform SAA. The results of the RO model were in accordance with the RO paradigm. This model indeed minimized the maximum cost for almost all cases. Table 5.2 shows that only occasionally another approach performed better regarding worst-case cost minimization. In general, the VRO is clearly present. However, RO is subject to a trade off between performance and conservatism. Therefore, it is also possible to generate solution with a good average cost. By selecting less restrictive budgets and lower variability levels one can control the protection against worst-case deviations, and thereby still retain a good average performance whilst not increasing the price of robustness too drastically (Bertsimas & Sim, 2004). Table 5.1 shows that with the proper parameter settings the average performance of RO is very promising.

We also fixed specific parameters to provide a more detailed analysis. We considered the cases where utilization is high or the coefficient of variation is low. These results are stated in Table 5.3 and Table 5.4, respectively.

Table 5.3: VMU, VSP, and VRO for the different methods (high utilization)

	VMU	VSP	VRO	p-value SAA difference
LP	0.00%	40.99%	21.90%	< 0.001***
Safety stock	27.62%	1.95%	3.82%	0.014*
RO	28.54%	0.64%	0.00%	< 0.001***
SAA	28.99%	0.00%	1.45%	
DRO	28.61%	0.55%	0.94%	0.003**

*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

For high levels of the capacity utilization, it turned out that the SAA approach bested all the other approaches. The VMU slightly reduced due to a reduction in capacity, which resulted in a more restrictive solution space for the approaches that cover the uncertainty which is inherent to the problem.

Table 5.4: VMU, VSP, and VRO for the different methods ($cv = 0.25$)

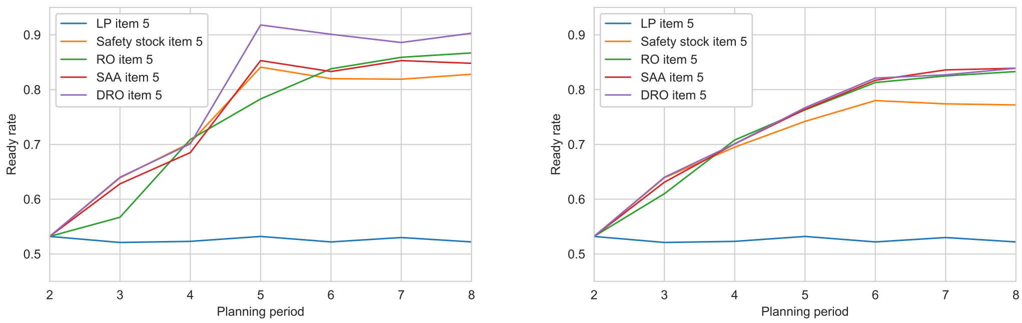
	VMU	VSP	VRO	p-value DRO difference
LP	0.00%	45.68%	32.42%	< 0.001***
Safety stock	31.66%	-0.45%	4.69%	< 0.001***
RO	32.18%	-1.19%	0.00%	< 0.001***
SAA	31.35%	0.00%	8.03%	< 0.001***
DRO	33.21%	-2.70%	2.82%	

*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Finally, Table 5.4 displays the aggregate results for the cases where $cv = 0.25$. For these situations the SAA model was outperformed by all the other models that model uncertainty explicitly. Distributionally robust optimization provided significantly stochastically smaller costs compared to all the other models. The deteriorated performance of SAA could be due to the fact that the scenario tree sampling procedure was more restrictive with regard to the generation of extreme scenarios. However, the DRO paradigm still protects against these outliers. Analogously, the RO approach provides a suitable buffer for this magnitude of variance.

5.2 Analysis of Solution Characteristics

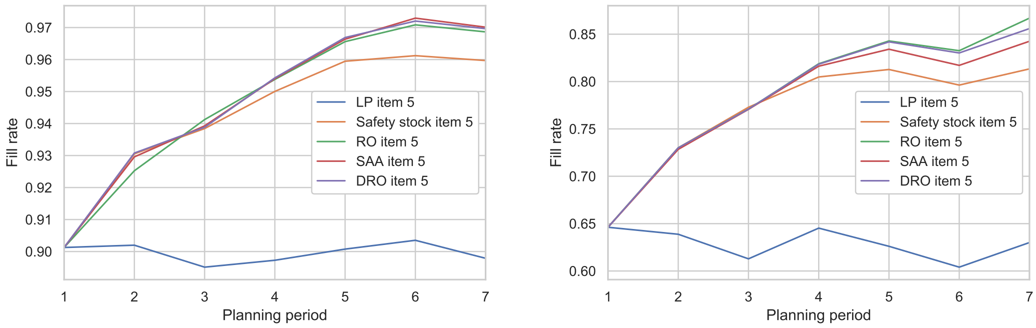
This section provides a more detailed description of the underlying planning behaviour of the solutions provided by the five different optimization approaches. First of all, Figures 5.1 and 5.2 shows the customer service levels that were attained.



(a) Ready rate low utilization (b) Ready rate high utilization

Figure 5.1: Ready rate of cases 4 and 16 ($cv = 0.25$ and symmetric supply chain)

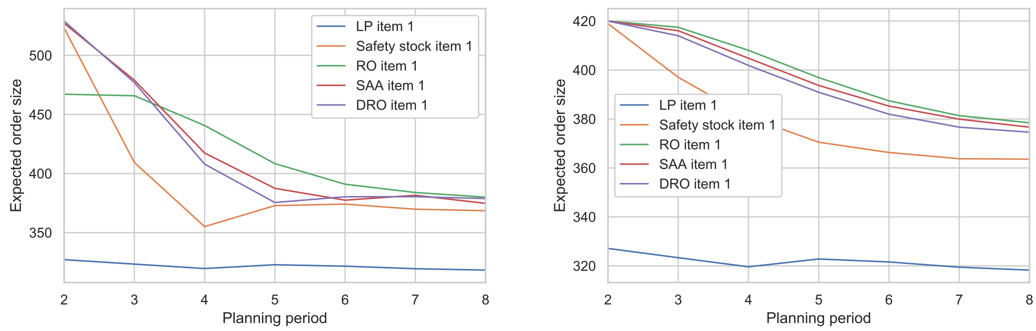
As expected, the service level performance of the deterministic LP was severely outperformed by the other methods. When increasing the utilization of the machines, the service level performance for most models deteriorated. The RO model retained a stable ready rate performance. It is noticeable that the target ready rate level (P_1-measure) for the safety stock procedure was not attained. This level was set to 0.91 in the final echelon stock points according to the newsvendor equations evaluated with the penalty/holding costs given in Section 4.2. This could be explained by the fact that we deal with a capacitated system and used a model that assumes infinite capacity for a heuristic procedure to determine proper safety stock levels.



(a) Fill rate low $cv = 0.25$ (b) Fill rate high $cv = 1$

Figure 5.2: Fill rates of cases 16 and 18 (high utilization and symmetric supply chain)

Figure 5.2 shows the reduction in fill rate performance (P_2-measure) when increasing the coefficient of variation. In the low variation case our three main models performed similarly. When the magnitude of the variance increased, the fill rate decreased. However, we see that the phenomena were in accordance to the optimization paradigms. RO provided a conservative safety stock level to buffer the uncertainty and outperformed the other methods with regard to this measure. The protection level of DRO was situated in between that of RO and SAA, which was the behaviour we expected.

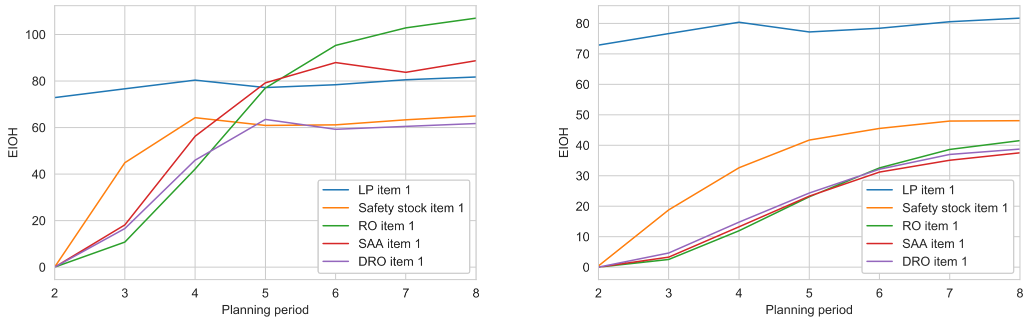


(a) $\mathbb{E}[OS]$ low utilization (b) $\mathbb{E}[OS]$ high utilization

Figure 5.3: Expected order size item 1 of cases 5 and 17 ($cv = 0.5$ and symmetric supply chain)

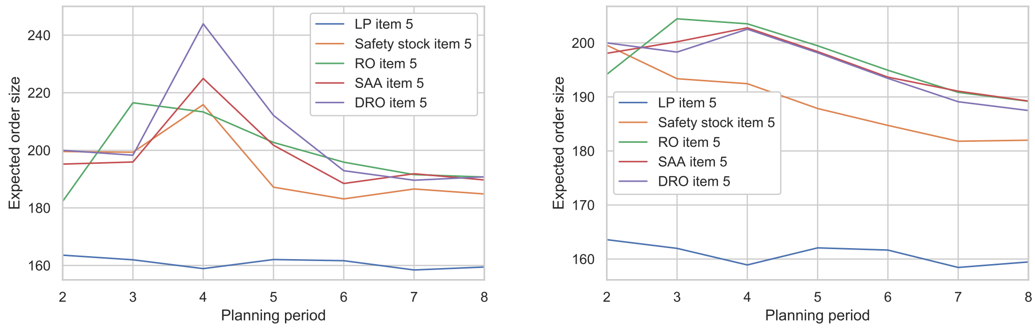
In the following figures, the expected order size and inventory on hand levels are displayed. The expected inventory on hand was measured at the end of a period when demand is fulfilled before the arrival of the products that were ordered one lead time ago. Figure 5.3 shows that the expected order size was greater than the sum of the expected demand in the first period, and that this level dropped down below the total expected demand in later periods. This had to do with the lost-sales assumption, which implied that the pipeline stock is unequal to the total expected demand. Hence, one builds up stock in the earlier periods to reduce lost sales, and needs to order less in the future. For higher levels of the utilization, the initial orders were smaller as there is less capacity at the final nodes to produce the end items. The on-hand stocks that were generated at PU1 are displayed in Figure 5.4. For low utilization levels RO built the largest safety stock for item 1. When utilization of the resources was increased, the safety stock model provided the highest inventory buffers at the root node.

A similar analysis was performed for the end items. In Figure 5.5a we noticed a peak in the order size at the fourth period, except for RO which, due to its conservatism, already



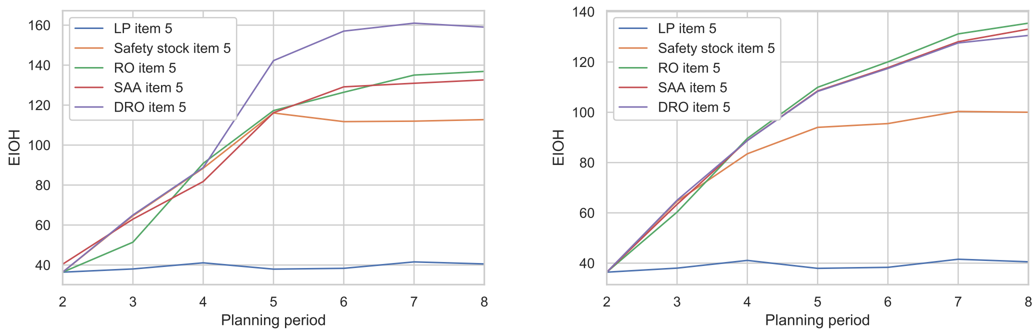
(a) $\mathbb{E}[IOH]$ item 1 low utilization (b) $\mathbb{E}[IOH]$ item 1 high utilization

Figure 5.4: Expected inventory on hand item 1 of cases 5 and 17 ($cv = 0.5$ and symmetric supply chain)



(a) $\mathbb{E}[OS]$ low utilization (b) $\mathbb{E}[OS]$ high utilization

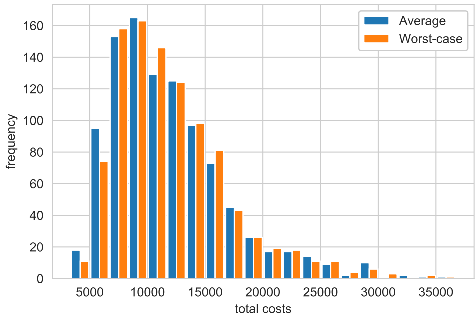
Figure 5.5: Expected order size item 5 of cases 5 and 17 ($cv = 0.5$ and symmetric supply chain)



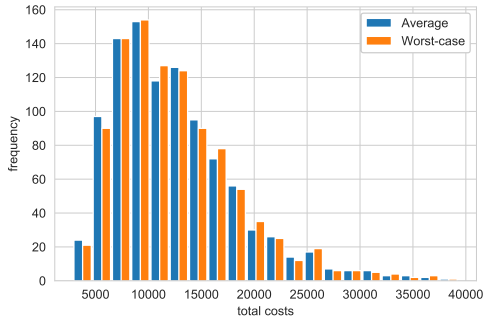
(a) $\mathbb{E}[IOH]$ item 5 low utilization (b) $\mathbb{E}[IOH]$ item 5 high utilization

Figure 5.6: Expected inventory on hand item 5 of cases 5 and 17 ($cv = 0.5$ and symmetric supply chain)

started ordering more one period earlier. This peak represented a accumulation in stock for future periods, to protect against high demand variation later on. Due to capacity restrictions, the system required stock accumulation in earlier periods to create a buffer. The inventory levels at the end of a planning period for item 5 are given by Figure 5.6. Distributionally robust optimization generated the highest inventory levels. The safety stock method created considerably less inventory.

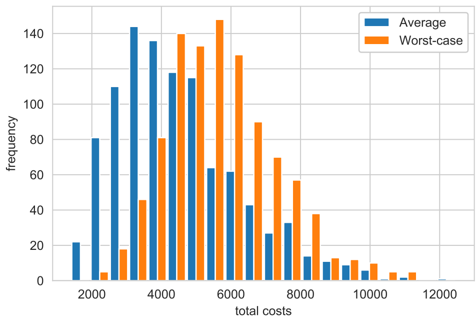


(a) DRO cost distribution low utilization

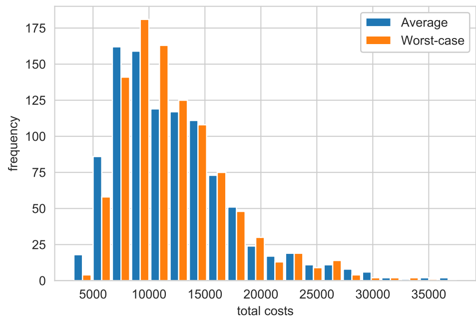


(b) DRO cost distribution high utilization

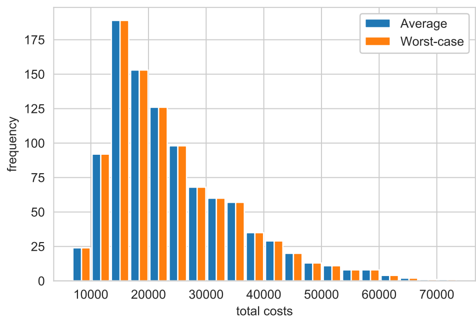
Figure 5.7: DRO total cost distribution of cases 2 and 14 ($cv = 0.5$ and asymmetric supply chain)



(a) RO cost distribution case 1



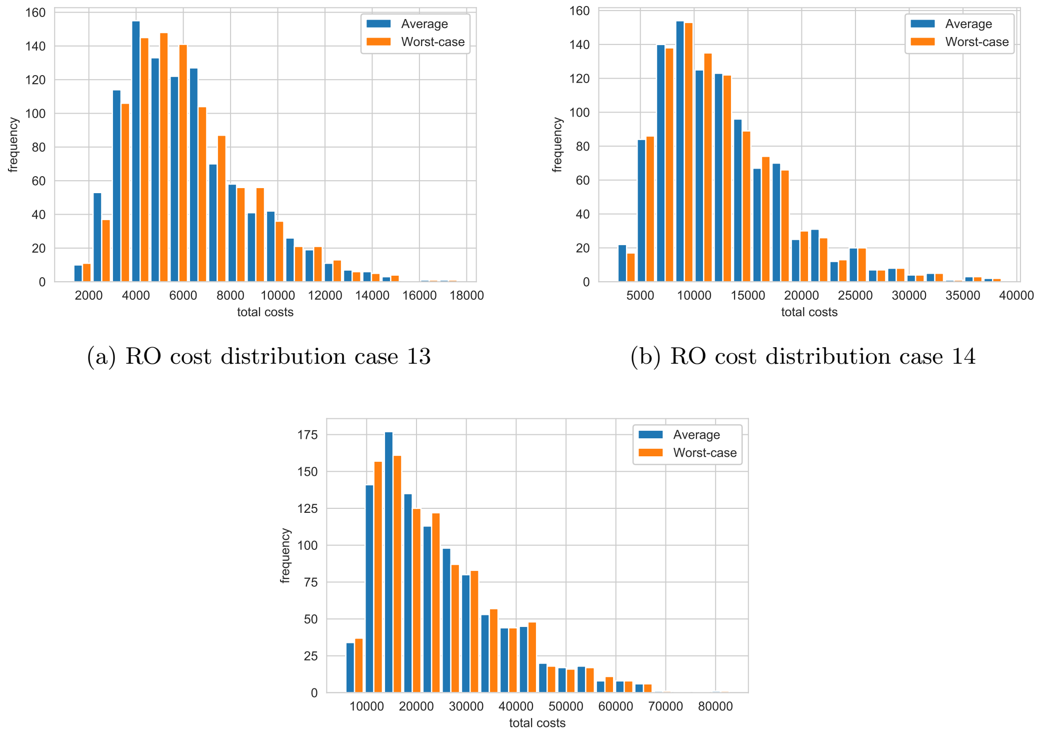
(b) RO cost distribution case 2



(c) RO cost distribution case 3

Figure 5.8: RO total cost distribution of low utilization cases (with asymmetric supply chain)

Both the DRO and RO models had an additional degree of freedom for tuning some optimization parameters. Hence, it is of interest to look at the difference between the solutions for the average and worst-case situation. Figure 5.7 displays the empirical distribution functions for the average and worst-case optimization of the DRO model. The DRO model seemed fairly insensitive regarding the choice of variability level. For most classes it turned out that the optimal value for the variability level was situated in the interval $\gamma \in [1.75, 2.5]$. However, the contrary holds for RO. For several classes the parameter settings and empirical distribution function of the total costs were significantly different. According to the histograms in Figure 5.8 their exists a discrepancy between the performance of the optimal average and worst-case minimization parameter settings. In particular, this holds for lower levels of the coefficient of variation. The worst-case oriented solution will overprotect to mitigate the maximum cost. However, when the amount of variation increased the two orientations would return similar results. This was due to the average case parameter setting also generating a conservative buffer against the high demand uncertainty. Finally, Figure 5.9 shows that the magnitude of the discrepancies reduced when increasing the utilization of the resources.



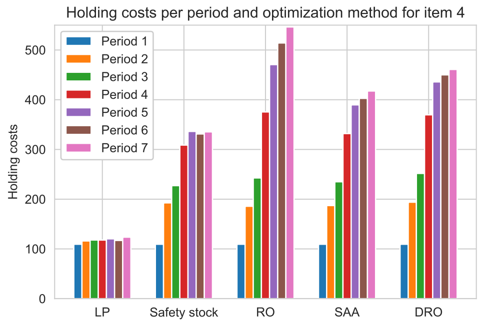
(a) RO cost distribution case 13

(b) RO cost distribution case 14

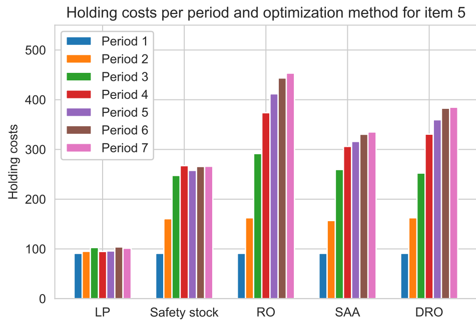
(c) RO cost distribution case 15

Figure 5.9: RO total cost distribution of high utilization cases (with asymmetric supply chain)

Finally, the holding/penalty costs were analyzed in greater detail. According to Figure 5.10, the total holding costs were distributed according to the general theory behind the optimization paradigms. Robust optimization incurred the highest holding costs to maintain the conservative inventory buffers. Distributionally robust optimization positioned itself in between RO and SAA. Hence, for these three approaches, the penalty costs were the highest for SAA and the lowest for RO. The safety stock approach had relatively high penalty costs, but significantly lower holding costs. Figure 5.12 provides the amount of lost sales per period for the different methods.

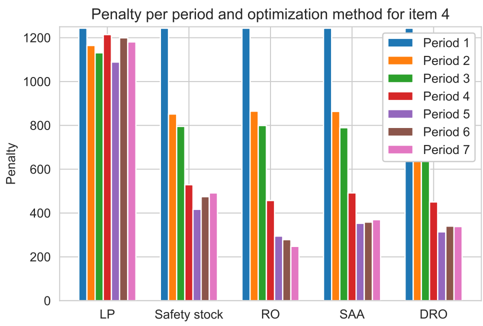


(a) Holding cost distribution over planning horizon for item 4

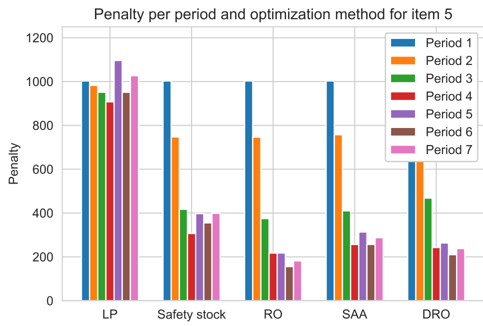


(b) Holding cost distribution over planning horizon for item 5

Figure 5.10: Average holding costs over planning horizon (case 8)

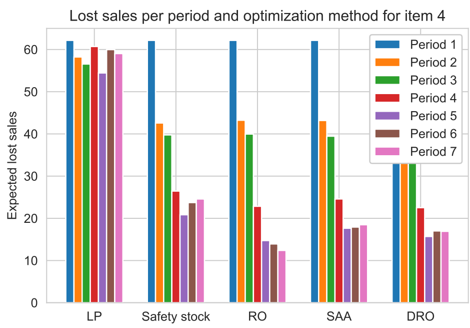


(a) Penalty cost distribution over planning horizon for item 4

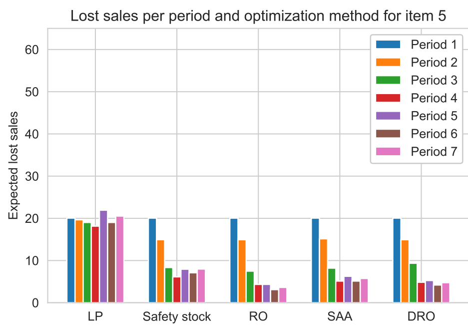


(b) Penalty cost distribution over planning horizon for item 5

Figure 5.11: Average penalty costs over planning horizon (case 8)



(a) Lost-sales distribution over planning horizon for item 4

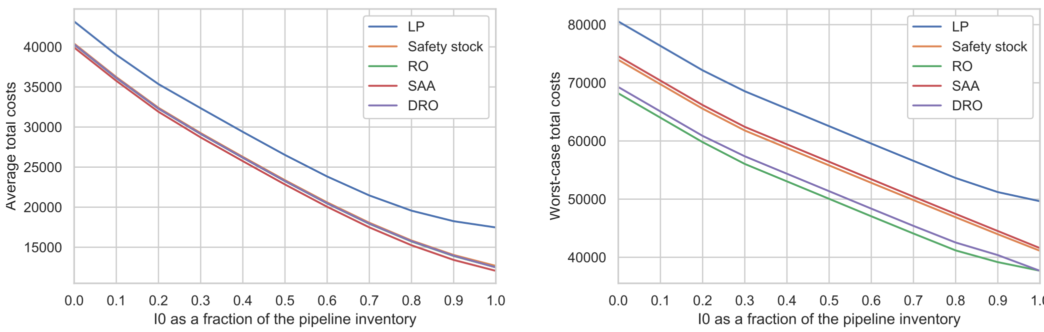


(b) Lost-sales distribution over planning horizon for item 5

Figure 5.12: Average lost sales over planning horizon (case 8)

5.3 Experiment Extension Results

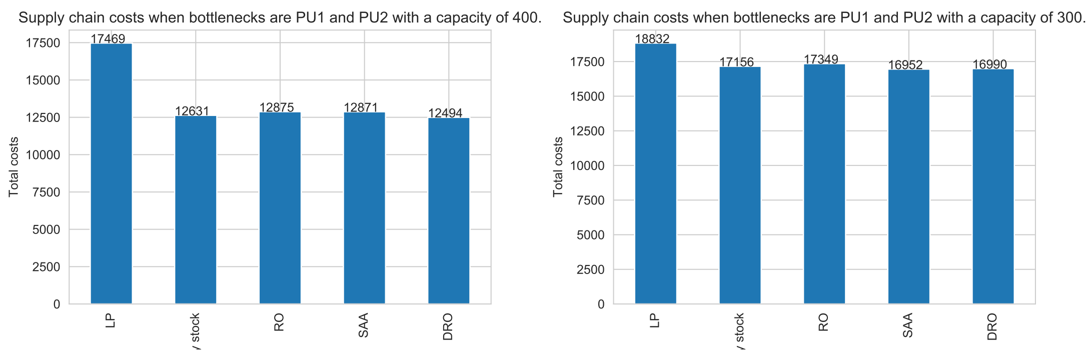
Finally, we looked at the sensitivity of our methods regarding the placement of the bottleneck and the amount of initial inventory. Figure 5.13 shows the sensitivity to initial inventory levels for case 8 of the experimental instances. The general results remain the same. The SAA approach had the best average performance, and the RO and DRO methods mitigated the worst-case scenario. However, the average/worst-case inventory costs were significantly affected by the initial stock levels I_0.
Shifting the bottleneck induced a remarkable result, which is stated in Figure 5.14. PU1 and PU2 were chosen as the bottlenecks and could only produce the total expected demand per period. The average performance of the different methodologies was almost indistinguishable (except for the deterministic LP). In general, all of the approaches keep only a limited amount of inventory at the upstream points, and place most of their stock at the end nodes. However, as the production of items 1, 2, and 3 was limited, all inventory was pushed downstream at a higher pace. As PU3 and PU4 had infinite capacity, the safety stock levels provided by ChainScope were probably close to optimal. Hence, the primarily inferior safety stock approach turned out to be quite useful in these cases. This model is the most computationally tractable technique, and therefore suffices as an excellent heuristic for these type of instances.



(a) Average costs with changing initial inventory level

(b) Worst-case costs with changing initial inventory level

Figure 5.13: Average/worst-case total cost when PU4 and PU5 are the bottlenecks



(a) Average costs with restricted capacity on PU1 and PU2 ($c_{1t} = c_{2t} = 400$)

(b) Average costs with restricted capacity on PU1 and PU2 ($c_{1t} = c_{2t} = 300$)

Figure 5.14: Average/worst-case total cost when PU4 and PU5 are the bottlenecks

Chapter 6

MOQ Model Extension

This chapter discusses the extension of our model which includes the binary setup decisions. We compared the performance of the different methods regarding the total cost of the system, but in the previous sections we neglected the binary decision variables and computation times. In this chapter, a small extension is provided which studies the computational tractability of the various methods, which were described in Chapter 3, on the full model stated in Section 1.3.1.

6.1 Methods

First of all, let us state the following theorem:

Theorem 6.1.1. *The deterministic MLCLSP with lost sales and MOQ constraints given by*

$$
\begin{aligned}
\min \quad & \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}[h_k I_{kt} + b_k L_{kt}] & \\
s.t. \quad & I_{k,t-1} + Q_{k,t-z_k} - \sum_{j\in\mathcal{S}_k} a_{kj}Q_{jt} = I_{k,t} & \forall k \in \mathcal{K}\backslash\mathcal{E},\ \forall t \in \mathcal{T} \\
& I_{e,t-1} + L_{et} + Q_{e,t-z_e} = I_{e,t} + d_{et} & \forall e \in \mathcal{E},\ \forall t \in \mathcal{T} \\
& \sum_{k\in\mathcal{K}_m} p_k Q_{kt} \leq c_{mt} & \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \\
& Q_{kt} \geq q_k^{min} Y_{kt} & \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \\
& Q_{kt} \leq (\sum_{m:k\in\mathcal{K}_m} c_{mt})Y_{kt} & \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \\
& Q_{kt}, I_{kt}, L_{kt} \geq 0 & \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \\
& Y_{kt} \in \{0,1\} & \forall k \in \mathcal{K}, \forall t \in \mathcal{T}
\end{aligned}
$$

is NP-hard.

Proof. See Appendix C. ■

Due to the result of this theorem, we can state that the problem is relatively hard to solve. Computation times could explode when the supply chains size grows. For each item and period combination a binary decision variable is introduced to model setup decisions. As solving this problem by regular B&B procedures increases computation times dramatically, we opt for a set of three heuristic solution methodologies which will be compared. These include:

1. Setting a time limit of 5 minutes for solving a planning model.
2. Changing the stopping criteria of Gurobi by reducing the tolerated relative optimality gap to 5%.

3. Implementing a relaxation heuristic.

The first option simply limits the computation time that Gurobi can invest in solving one specific planning model. When $T = 6$ and $H = 12$, we solve 7 optimization problems in total. Hence, Gurobi has 35 minutes to optimize these models. This excludes the time Gurobi needs to construct/update the underlying model objects.
The second option is also rather straightforward. By introducing some slack to the (relative) gap between the upper and lower bound of the B&B algorithm, we reduce the time invested by Gurobi in proving that the current solutions is optimal. Consequently, the incumbent solution might not be optimal, but the solution time is reduced significantly while we are able to bound the decay in solution quality.
Lastly, relaxation heuristics offer another opportunity to reduce computation times. Our heuristic encompasses a linear relaxation of the binary setup variables. All binary decision variables for periods $t > 1$ will be relaxed. Hence, only concrete production decisions are made in the current period. The flexibility of our production-inventory system enables this option.

6.2 Experimental Setup

For the comparison of these methods we generated a new experimental test bed. We took the setup of class 8 from the original experiments. We constructed supply chains based on a binary tree. Figure 6.1 shows an example of the supply chain configuration. We generated three different supply chains with different numbers of items K: 15, 63, and 127 items. Each pair of descendants has a shared resource. Therefore, $M = \frac{K}{2} + 1$. Only the end nodes were subject to uncertain demand. Each resource at the end of the chain has capacity $c_{mt} = \frac{\sum_{\mathcal{E}_m} \mathbb{E}[d_{et}]}{0.85}$, where $\mathcal{E}_m$ is the set of end items that had required a transformation on machine m. The other resources had ample capacity.

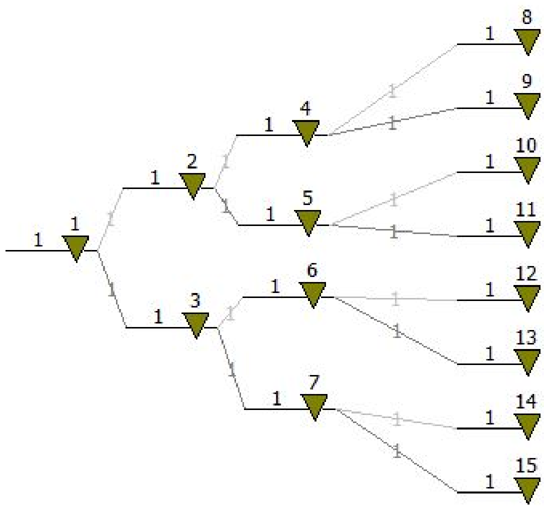


Figure 6.1: Supply chain configuration for MOQ experiments.

For each of these supply chains the time horizon was set to $H = 12$ with a planning horizon $T = 6$. The other input parameters were controlled by fixing them in the following way:

1. Expected end-item demand: $\mathbb{E}[\mathbf{d}] = \begin{bmatrix} 300 \\ 300 \\ \vdots \\ 100 \\ 100 \end{bmatrix}$
2. The holding costs were chosen incremental on the interval $[0.25, 2.5]$. More downstream goods or low demand items incurred higher holding costs.
3. Penalty costs: $b_e = 10h_e$, $\forall e \in \mathcal{E}$
4. Processing times: $p_k = 1$, $\forall k \in \mathcal{K}$
5. Capacity non-bottleneck PUs: $c_{mt} = 2\sum_{e \in \mathcal{E}_m} \mathbb{E}[d_{et}]$, $\forall t \in \mathcal{T}$
6. Planned lead times: $z_k = 1$, $\forall k \in \mathcal{K}$
7. Internal demands: $a_{kj} = 1$, $\forall j \in \mathcal{S}_k$, $\forall k \in \mathcal{K}$

Finally, the MOQ of k, q_k^{min}, was set equal to the sum of the expected demands of all end items for which k serves as a component. For an end item e, q_e^{min} was set to its respective expected demand. A small sensitivity analysis was performed regarding the MOQ parameter. For the different numbers of items we measured the costs and computation times of the different optimization methods by executing a Monte Carlo simulation procedure with 100 iterations. However, this time we restricted the variability level and budget of uncertainty to the values that were optimal for case 8 of the previous test bed. We took $\gamma = 2$ and $\Gamma(t) = 0.5t + 0.5$. The branching factor was set to 6, with a threshold at period 3. This resulted in a scenario tree with 145 nodes.

6.3 Results

All the computational experiments were implemented in the Python programming language and the models were solved using Gurobi 8.1.1 on a 12-core Intel(R) Xeon(R) X5675 3.07GHz processor. Tables 6.1 and 6.2 show the results of our computational experiments. The stated computation time presents the average amount of time required for solving the seven distinct planning problems that form one iteration of the Monte Carlo simulation.

Table 6.1: Costs and computation time results of MOQ experiments, 15 and 63 items.

		15 items			63 items		
		Average cost	σ	Computation time (sec)	Average cost	σ	Computation time (sec)
Safety stock	Default	71850	9148	0.46	171139	20161	3.62
	5% optimality gap	74473	8788	0.26	173645	20438	1.48
	Relaxation heuristic	91496	11098	0.11	375976	31549	0.94
RO	Default	64104	8343	5.35	162636	19467	80.42
	5% optimality gap	75847	8589	1.62	236303	22284	15.32
	Relaxation heuristic	76008	8650	1.88	247869	22685	12.89
SAA	Default	63636	8389	155.95	149085	25840	-
	5% optimality gap	65537	8924	23.68	149085	25840	-
	Relaxation heuristic	68296	8325	60.96	152090	22832	-
DRO	Default	63934	8750	17.45	156161	21231	166.75
	5% optimality gap	63708	8610	12.21	158418	21667	53.94
	Relaxation heuristic	63277	8151	10.72	176372	19859	52.47

The first table displays the results for 15 and 63 items. It turns out that the SAA approach is significantly slower than the other methods. Furthermore, the acceptance of a 5% optimality gap and the relaxation heuristic deteriorated the cost performance of the models, but these modifications also resulted in reduced computation times. However, the relaxation heuristic was severely outperformed regarding costs. Note that the SAA method already needed all available computation time for the cases were the number of items is greater than 15. The safety stock approach delivered relatively poor quality solution. This is related to the lot sizing decisions induced by the MOQ. Distributionally robust optimization provides the best cost performance, but this method requires more computation time than the safety stock heuristic and RO.
Table 6.2 displays another remarkable result. The SAA approach required all available computation time, but the average costs increase considerably compared to RO and DRO. The former method was not able to generate good solutions in the available running time. The DRO method provided the best solutions in acceptable computation times.

Table 6.2: Cost and computation time results of MOQ experiments, 127 items.

		127 items		
		Average cost	σ	Computation time (sec)
Safety stock	Default	783594	42480	189.45
	5% optimality gap	790122	43215	4.49
	Relaxation heuristic	986207	51902	3.10
RO	Default	387088	35135	706.29
	5% optimality gap	571944	46355	53.35
	Relaxation heuristic	607666	38778	40.29
SAA	Default	592112	111454	-
	5% optimality gap	592112	111454	-
	Relaxation heuristic	644218	97745	-
DRO	Default	343687	32017	898.71
	5% optimality gap	381732	32464	162.45
	Relaxation heuristic	443804	38828	148.43

Figure 6.2 shows the results of a sensitivity analysis regarding the MOQ parameter q_k^{min}. The low, medium, and high value refer to 0.25, 0.5, and 1 times the excepted demand as minimum order quantity, respectively. It turns out that the MOQ does not affect the performance of the solutions for the first two values. However, when setting the MOQ equal to the expected demand the system performance reduced considerably. In particular, the induced lot size in the latter situation greatly impairs the safety stock heuristic. This method does not account for lot sizing, and therefore it was strongly affected by the greater MOQ sizes. Notice that the safety stock heuristic performed quite wel for small/moderate sizes of the MOQ.

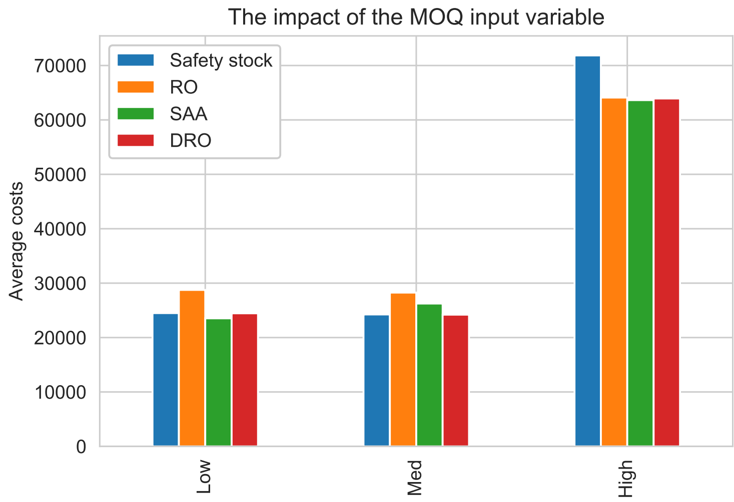


Figure 6.2: Sensitivity analysis on the MOQ parameter q^{min}

Chapter 7

Discussion

In conclusion, this section presents a summary of the most relevant results and their implications. Furthermore, we discuss the limitations of our research and provide directions for future research.

7.1 Conclusions and Managerial Insights

This work presented several optimization approaches to tackle the MLCLSP with lost-sales and uncertain demand. These methods include a deterministic planning model, an LP model with safety stocks, robust optimization, sample average approximation, and distributionally robust optimization. Chapter 5 describes the results of our research.
First of all, we emphasize the relevance of our research for supply chain planning problems in industry. In essence, the problem defined is the stochastic MLCLSP. Our model incorporates a significant set of issues that occur in practice. Furthermore, we are the first to construct models that are able to solve lost-sales inventory problems by using robust optimization techniques. The derivations of the RO and DRO models for the lost-sales case are discussed in Chapter 3. By using an expression for the cumulative lost sales over a planning horizon, which is based on expression (3.1) in the work of Van Donselaar, De Kok, and Rutten (1996), we proved convexity of our objective function and derived the exact robust counterparts that could be used for the model in a rolling horizon context. Our research has provided a set of heuristic optimization models that are appropriate for solving real-life planning problems in terms of the S-MLCLSP with lost-sales. These MP-based methods are implementable in contemporary APS modules.
In general, our research shows the value of modeling uncertainty of demand explicitly when solving an S-MLCLSP problem. On average, the relative improvement over the deterministic case hit a staggering 30%. To assess the advantages/disadvantages of the other optimization methods, we constructed an extensive Monte Carlo simulation study. Based on the results of the study, it can be concluded that the underlying planning behaviour of the different methods conforms to the general phenomena that are observed for these specific optimization paradigms. Robust optimization tends to expand the inventory buffer to mitigate the risk of high penalty costs. Sample average approximation has the propensity to focus on the expected costs, and distributionally robust optimization possesses the aptitude to protect against the worst-case expected behaviour. Hence, in general distributionally robust optimization maintains higher inventory levels than SAA, and therefore the former provides higher customer service levels.
Based on the results of Chapter 5, DRO turns out to be the optimization method with the best overall results. On average, DRO outperforms all other methods. In particular, DRO shows significant dominance over the other methods when applied to instances with low demand variability. Furthermore, the former method shares a beneficial quality with robust optimization as both are distribution-free methods. Hence, we only need limited information on the demand distribution, which consists of the first two moments and bounds on the support. This

contrasts with the SAA method, which needs detailed information on the demand distribution to generate the scenario tree. Distributionally robust optimization utilizes this form of insensitivity to outperform sample average approximation. The worst-case performance of DRO is inferior to regular robust optimization. However, the performance of DRO is less susceptible to the initialization values of its optimization input parameter: the variability level. The robust optimization methods seems more sensitive for parameter choices regarding the budget of uncertainty and the variability level. Therefore, the former technique seems to be more pragmatic. Often in practice, there is no possibility to simulate the planning horizon an indefinite amount of times. Hence, it is necessary to initialize the optimization method with the proper parameter values directly.
Another important finding is the effect of the utilization and coefficient of variation. It turns out that both performance and solution characteristics of the optimization methods are affected by these two inputs. The value of modeling the uncertainty reduces when resource capacity becomes more restrictive. However, the main three methods discussed in this report outperform the safety stock heuristic by a greater margin when utilization becomes higher. This makes perfect sense as the safety stock heuristic uses a model with incompatible modeling assumptions. Therefore, the computed safety stock levels are inadequate to optimize the costs of the system. The coefficient of variation plays an important role in the determination of the budget of uncertainty and variation level for the RO approach. When the magnitude of variance increases, the RO approach seems more robust regarding a proper parameter choice to both perform well on average and mitigate the worst-case risks.
The additional experiments show that the safety stock procedure that we developed is a suitable method when the bottleneck is placed upstream in the supply chain. The other methods show no significant improvements in this case. Furthermore, the safety stock procedure has less of a computational burden, which implies that this method can manage bigger supply chain structures with a greater number of items.
The experiments performed on the MOQ variant also provided us with valuable insights. First of all, it turns out that the performance of the SAA approach deteriorates considerably when the number of items increases. This method is unable to provide provable optimal results to the MIP model. Moreover, the incumbent solution that is returned is inferior to the solutions that RO and DRO provide. Distributionally robust optimization seems to outperform all the other methods. However, note that we only scaled the problem in the amount of items. Increasing the number of planning periods would probably affect the cost/computation time performance of the latter method considerably as well.
Lastly, to conclude, we like to refer the interested reader to Appendix D. In this section we describe the application of our methods on a real-life business case in process industry. The MOQ variant of the S-MLCLSP was applied with as input parameters the master data of the company concerned. The methods developed in our research significantly outperformed the current situation, which used safety stock levels that were optimized by means of simulation based optimization. For this case we used forecasts instead of actual demand data. Nevertheless, our methods provide a significant improvement and seem rather robust to this deficiency in information. This proof-of-concept case study shows the practical relevance of our methods.

7.2 Limitations and Future Research

From a practical point of view, one of this research's major limitations is the direct insertion of perfect information regarding the moments of the demand distribution. To mimic realistic SCOP problems, one should consider using the actual demand realizations to provide estimates for these moments. Subsequently, the forecast (error) of the demand can be used to generate the scenario tree when using SAA or to construct the RO/DRO models. The effect of these changes on the performance of the different methods is an interesting topic for future endeavors in this area.
Another topic of interest that should be discussed is the computational tractability as a function of the length of the planning horizon under consideration. We studied the effect of increas-

ing the size of the supply chain on the computation times. However, the input parameter T could have a greater impact on this metric. Note that for RO, the number of constraints is exponential in the parameter T and also depends on the specified budget of uncertainty. For the case $\Gamma(t) = t$, i.e. Soyster's model, the amount of constraints is of the order $\mathcal{O}(2^T)$. The same issue arises for the SAA method. The total size of the scenario tree is $\prod_{t=1}^{T} a(t) + 1$. Hence, the size of the optimization model also explodes exponentially in the parameter T. Moreover, DRO generates a model with $\mathcal{O}(3^T)$ constraints. Hence, this model also inflates with an exponential rate regarding the length of the planning horizon. There exist several methods to reduce this order of complexity to $\mathcal{O}(2^T)$, or even $\mathcal{O}(T)$, by aggregating several demand parameters (Postek et al., 2018). These upgrades are an interesting topic to explore. One should expect an improvement in computational tractability, but the cost performance would potentially decrease.

Our research used several heuristics to generate feasible solutions to the MOQ variant of our problem. However, one could consider applying more advanced techniques exploiting the structures of the RO, SAA, and DRO models. These models comprise of either vertices, scenarios, or permutations, respectively. As such, these models can be decomposed and methods such as Bender's Decomposition could prove to be useful. Additionally, selection algorithms could be developed which search for vertices/scenarios/permutations that strongly affect the solution and discard irrelevant ones. Finally, one could consider reformulations of the MLCLSP, such as the SPLP, to tighten the linear relaxation of the problem, and hence decrease the time needed to solve the MIP model by applying B&B.

Our experimental design fixed several input parameters. Therefore, future research could expand on our work by measuring the impact of these parameters on system performance. E.g., one could vary the penalty costs b_k and/or the planning horizon T. In a rolling horizon context, the planning horizon could significantly affect cost performance due to planning nervousness. An interesting followup question is which of the discussed methods is most robust against this nervousness. Furthermore, our experimental designs only considered divergent supply chains. In the future, several other supply chain configurations should be modeled and solved by the discussed methods to provide a test of validity.

Another topic to consider is testing our models against several different demand distributions. Sampling demand from strongly skewed distributions could potentially lead to significant deterioration of the RO models, which assume symmetrical distributions.

Finally, one could consider extensions of the methods that we applied in this thesis. One of these options is adjustable robust optimization. According to the work of (Gorissen et al., 2015), this method considers a less conservative and more flexible solution space. Therefore, this method could prove to be more suitable when optimizing average performance. Analogously, the SAA method can be extended by considering the Conditional Value-at-Risk. This metric can be incorporated into the target function of the SAA method. Introducing this metric could result in better worst-case performance by reducing tail risks.

References

Absi, N., & Kedad-Sidhoum, S. (2009, 11). The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs. *Computers & Operations Research*, *36*(11), 2926–2936. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054809000185

Alem, D., Curcio, E., Amorim, P., & Almada-Lobo, B. (2018). A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers and Operations Research*, *90*, 125–141.

Alfieri, A., Brandimarte, P., & D'Orazio, S. (2002, 1). LP-based heuristics for the capacitated lot-sizing problem: The interaction of model formulation and solution algorithm. *International Journal of Production Research*, *40*(2), 441–458. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/00207540110081461

Armentano, V. A., Franca, P. M., & De Toledo, F. M. B. (1999). A network flow model for the capacitated lot-sizing problem. *Omega*, *27*(2), 275–284.

Balasubramanian, J., & Grossmann, I. E. (2004). Approximation to Multistage Stochastic Optimization in Multiperiod Batch Plant Scheduling under Demand Uncertainty. *Ind. Eng. Chem. Res.*, *43*(14), 3695–3713. Retrieved from https://pubs.acs.org/doi/full/10.1021/ie030308+

Barany, I., & Van Roy, T. J. (1984). Strong formulations for multi-item capacitated lot sizing. *Management Science*, *30*(10), 1255–1261.

Ben-Tal, A., Boaz, G., & Shimrit, S. (2009, 12). Robust multi-echelon multi-period inventory control. *European Journal of Operational Research*, *199*(3), 922–935. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221709002112

Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust Optimization.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.587.8223&rep=rep1&type=pdf

Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004, 3). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, *99*(2), 351–376. Retrieved from http://link.springer.com/10.1007/s10107-003-0454-y

Beraldi, P., Ghiani, G., Grieco, A., & Guerriero, E. (2006, 3). Fix and Relax Heuristic for a Stochastic Lot-Sizing Problem. *Computational Optimization and Applications*, *33*(2-3), 303–318. Retrieved from http://link.springer.com/10.1007/s10589-005-3055-2

Beraldi, P., Ghiani, G., Grieco, A., & Guerriero, E. (2008, 11). Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, *35*(11), 3644–3656. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054807000858

Bertrand, J. W. M., & Fransoo, J. C. (2002, 2). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, *22*(2), 241–264. Retrieved from https://www.emeraldinsight.com/doi/10.1108/01443570210414338

Bertrand, J. W. M., Wortmann, J. C., & Wijngaard, J. (1990). *Production control: a structural and design oriented approach.* Amsterdam: Elsevier.

Bertsimas, D., & Sim, M. (2004, 2). The Price of Robustness. *Operations Research*, *52*(1), 35–53. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.1030.0065

Bertsimas, D., & Thiele, A. (2006, 2). A Robust Optimization Approach to Inventory Theory. *Operations Research*, *54*(1), 150–168. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.1050.0238

Bienstock, D., & Özbay, N. (2008, 5). Computing robust basestock levels. *Discrete Optimization*, *5*(2), 389–414. Retrieved from https://www.sciencedirect.com/science/article/pii/S1572528607000382

Billington, P. J., McClain, J. O., & Thomas, L. J. (1983, 10). Mathematical Programming Approaches to Capacity-Constrained MRP Systems: Review, Formulation and Problem Reduction. *Management Science*, *29*(10), 1126–1141. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/mnsc.29.10.1126

Bitran, G. R., & Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, *28*(10), 1174–1186.

Boctor, F. F., & Poulin, P. (2005, 7). Heuristics for the N-product, M-stage, economic lot sizing and scheduling problem with dynamic demand. *International Journal of Production Research*, *43*(13), 2809–2828. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/00207540500057589

Bourjolly, J.-M., Ding, K., Gopalakrishnan, M., Gramani, M. C., & Mohan, S. (2001). *On a Tactical/Operational Production Planning Problem in Supply Chain Management: Balancing Inventory and Setup Costs* (Tech. Rep.). Retrieved from https://pdfs.semanticscholar.org/45e3/173580710842c53f7246d7fd9fcfc71da882.pdf

Boyd, S. P., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge: Cambridge University Press. Retrieved from http://ebooks.cambridge.org/ref/id/CBO9780511804441

Brandimarte, P. (2006). Multi-item capacitated lot-sizing with demand uncertainty. *International Journal of Production Research*, *44*(15), 2997–3022.

Bruggeman, W., & Jahnke, H. (2000). The discrete lot-sizing and scheduling problem: complexity and modification for batch availability. *European Journal of Operational Research*, *124*, 511–528.

Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010, 4). Dynamic capacitated lot-sizing problems: A classification and review of solution approaches. *OR Spectrum*, *32*(2), 231–261. Retrieved from http://link.springer.com/10.1007/s00291-008-0150-7

Cattrysse, D., Maes, J., & Van Wassenhove, L. N. (1990). Set partitioning and column generation heuristics for capacitated dynamic lot sizing. *European Journal of Operational Research*, *46*(1), 38–47.

Clark, A. R., & Armentano, V. A. (1995, 10). A Heuristic for a Resource-capacitated Multi-stage Lot-sizing Problem with Lead Times. *Journal of the Operational Research Society*, *46*(10), 1208–1222. Retrieved from https://www.tandfonline.com/doi/full/10.1057/jors.1995.169

Curcio, E., Amorim, P., Zhang, Q., & Almada-Lobo, B. (2018). Adaptation and approximate strategies for solving the lot-sizing and scheduling problem under multistage demand uncertainty. *International Journal of Production Economics*, *202*(August 2017), 81–96. Retrieved from https://doi.org/10.1016/j.ijpe.2018.04.012

De Kok, A. G. (2018). Inventory Management: Modelling Real-life Supply Chains and Empirical Validity. *Foundations and Trends in Technology, Information and Operations Management*, *11*(4), 343–437.

De Kok, A. G., & Fransoo, J. C. (2003). Planning Supply Chain Operations: Definition and Comparison of Planning Concepts. *Handbooks in Operations Research and Management Science*, *11*(C), 597–675.

De Kok, A. G., Grob, C., Laumanns, M., Minner, S., Rambau, J., & Schade, K. (2018). A typology and literature review on stochastic multi-echelon inventory models. *European Journal of Operational Research*, *269*(3), 955–983. Retrieved from https://doi.org/10.1016/j.ejor.2018.02.047

Diks, E. B., & De Kok, A. G. (1998). Computational results for the control of a divergent N-echelon inventory system. *European Journal of Operational Research*, *111*, 75–97.

Dillenberger, C., Escudero, L. F., Wollensak, A., & Zhang, W. (1994, 6). On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, *75*(2), 275–286. Retrieved from https://www.sciencedirect.com/science/article/pii/0377221794900744

Drexl, A., & Haasse, K. (1995). Proportional lot sizing and scheduling. *International Journal of Production Economics*, *40*, 73–87.

Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling-survey and extensions. *European Journal of Operational Research*, *99*(2), 511–528.

Eisenhut, P. S. (1975, 6). A Dynamic Lot Sizing Algorithm with Capacity Constraints. *A I I E Transactions*, *7*(2), 170–176. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/05695557508974999

Eppen, G. D., & Martin, R. K. (1987). Solving multi-item capacitated lot sizing problems using variable redefinition. *Operations Research*, *35*(6), 832–848.

Erenguc, S. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production / distribution planning in supply chains : An invited review. *European Journal of Operational Research*, *115*(2), 219–236.

Federgruen, A., & Tzur, M. A. (1991). A simple forward algorithm to solve general dynamic lot sizing models with n perios in O(n log n) or O(n) time. *Management Science*, *37*(8), 909–925.

Fleischmann, B., & Meyr, H. (1997). The general lot sizing and scheduling problem. *OR Spectrum*, *19*(1), 11–21.

Florian, M., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1980). Determenistic production planning algorithms and complexity. *Management Science*, *26*(7), 669–679.

Fukuda, K., & Prodon, A. (2012). Double description method revisited..

Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, *235*(3), 471–483.

Gallego, G., & Shaw, D. X. (1997). Complexity of the ELSP with general cyclic schedules. *IEEE Transactions*, *29*, 109–113.

Glasserman, P., & Tayur, S. (1995, 2). Sensitivity Analysis for Base-Stock Levels in Multiechelon Production-Inventory Systems. *Management Science*, *41*(2), 263–281. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/mnsc.41.2.263

Glover, F. (1986, 1). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549. Retrieved from https://www.sciencedirect.com/science/article/pii/0305054886900481

Gorissen, B. L., & Den Hertog, D. (2013, 5). Robust counterparts of inequalities containing sums of maxima of linear functions. *European Journal of Operational Research*, *227*(1), 30–43. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221712007345https://arxiv.org/pdf/1501.02656.pdf

Gorissen, B. L., Yanıkoglu, I., & Den Hertog, D. (2015, 6). A practical guide to robust optimization. *Omega*, *53*, 124–137. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305048314001698

Guillaume, R., Thierry, C., & Zieliński, P. (2017). Robust material requirement planning with cumulative demand under uncertainty. *International Journal of Production Research*, *55*(22), 6824–6845.

Hansen, P., & Mladenović, N. (2001, 5). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, *130*(3), 449–467. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221700001004

Harris, F. W. (1913). How many parts to make at once. *The Magazine of Management*, *10*(2), 135–136.

Helber, S. (1995, 3). Lot sizing in capacitated production planning and control systems. *Operations-Research-Spektrum*, *17*(1), 5–18. Retrieved from http://link.springer.com/10.1007/BF01719725

Hopp, W., & Spearman, M. (2000). *Factory Physics* (Second ed. ed.). Boston: Irwin McGraw-Hill.

Hu, Z., & Hu, G. (2018). A multi-stage stochastic programming for lot-sizing and scheduling under demand uncertainty. *Computers and Industrial Engineering*, *119*(August 2017), 157–166.

Janakiraman, G., & Muckstadt, J. A. (2009, 12). A Decomposition Approach for a Class of Capacitated Serial Systems. *Operations Research*, *57*(6), 1384–1393. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.1080.0680

Karaarslan, A. G., Kiesmüller, G. P., & De Kok, A. G. (2013, 6). Analysis of an assemble-to-order system with different review periods. *International Journal of Production Economics*, *143*(2), 335–341. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925527311004932

Karimi, B., Fatemi Ghomi, S. M. T., & Wilson, J. M. (2003). The capacitated lot sizing problem: A review of models and algorithms. *Omega*, *31*(5), 365–378.

Karni, R., & Roll, Y. (1982). A heuristic algorithm for the multi-item lot sizing problem with capacity constraints. *IEEE Transactions1*, *14*(4), 249–259.

Kirca, O., & Kokten, M. (1994). A new heuristic approach for the multi-item dynamic lot sizing problem. *European Journal of Operational Research*, *75*(2), 332–341.

Kirkpatrick, S. (1984, 3). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, *34*(5-6), 975–986. Retrieved from http://link.springer.com/10.1007/BF01009452

Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002, 1). The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal on Optimization*, *12*(2), 479–502. Retrieved from http://epubs.siam.org/doi/10.1137/S1052623499363220

Kohler-Gudum, C. K., & De Kok, A. G. (2002). *A safety stock adjustment procedure to enable target service levels in simulation of generic inventory systems.* Eindhoven.

Kuik, R., & Salomon, M. (1990, 3). Multi-level lot-sizing problem: Evaluation of a simulated-annealing heuristic. *European Journal of Operational Research*, *45*(1), 25–37. Retrieved from https://www.sciencedirect.com/science/article/pii/0377221790901533

Leung, J. M. Y., Magnanti, T. L., & Vachani, R. (1989). Facets and algorithms for capacitated lot sizing. *Mathematical Programming*, *45*, 331–359.

Maes, J., & McClain, J. O. (1991). Multilevel capacitated lot sizing complexity and LP-based heuristics. *European Journal of Operational Research*, *53*(2), 131–148.

Maes, J., & Van Wassenhove, L. N. (1986). A simple heuristic for the multi-item single level capacitated lot sizing problem. *Operations Research Letters*, *4*(6), 265–273.

Maes, J., & Van Wassenhove, L. N. (1988). Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review. *Journal of the Operational Research Society*, *39*(11), 991–1004.

Melamed, M., Ben-Tal, A., & Golany, B. (2016, 4). On the average performance of the adjustable RO and its use as an offline tool for multi-period production planning under uncertainty. *Computational Management Science*, *13*(2), 293–315. Retrieved from http://link.springer.com/10.1007/s10287-016-0250-9

Millar, H. H., & Yang, M. (1994, 2). Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. *International Journal of Production Economics*, *34*(1), 1–15. Retrieved from https://www.sciencedirect.com/science/article/pii/0925527394900426

Minoux, M. (2018). Robust and stochastic multistage optimisation under Markovian uncertainty with applications to production/inventory problems. *International Journal of Production Research*, *56*(1-2), 565–583.

Mitroff, I. I., Betz, F., Pondy, L. R., & Sagasti, F. (1974, 5). On Managing Science in the Systems Age: Two Schemas for the Study of Science as a Whole Systems Phenomenon. *Interfaces*, *4*(3), 46–58. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/inte.4.3.46

Özdamar, L., & Barbarosoğlu, G. (1999, 8). Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem. *Journal of the Operational Research Society*, *50*(8), 810–825. Retrieved from https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2600773

Özdamar, L., & Barbarosoglu, G. (2000, 12). An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. *International Journal of Production Economics*, *68*(3), 319–331. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925527399001000

Özdamar, L., & Bozyel, M. A. (2000). The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions*, *32*(11), 1043–1057. Retrieved from http://link.springer.com/10.1023/A:1013733024060

Peidro, D., Mula, J., Poler, R., & Lario, F.-C. (2009). Quantitative models for supply chain planning under uncertainty: a review. *The International Journal of Advanced Manufacturing Technology*, *43*(3-4), 400–420.

Pitakaso, R., Almeder, C., Doerner, K. F., & Hartl, R. F. (2005). *Combining Population-Based and Exact Methods for Multi-Level Capacitated Lot Sizing Problems* (Vol. 14; Tech. Rep. No. 00). Retrieved from http://www.ubu.ac.th/~pitakaso/1302450/Related/Pit_2006_286.pdf

Postek, K., Ben-Tal, A., Den Hertog, D., & Melenberg, B. (2018, 6). Robust Optimization with Ambiguous Stochastic Constraints Under Mean and Dispersion Information. *Operations Research*, *66*(3), 814–833. Retrieved from http://pubsonline.informs.org/doi/10.1287/opre.2017.1688

Robert Jacobs, F., & Khumawala, B. M. (1982, 1). Multi-level lot sizing in material requirements planning: an empirical investigation. *Computers & Operations Research*, *9*(2), 139–144. Retrieved from https://www.sciencedirect.com/science/article/pii/0305054882900120

Rosling, K. (1989, 8). Optimal Inventory Policies for Assembly Systems Under Random Demands. *Operations Research*, *37*(4), 565–579. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.37.4.565

Sambasivan, M. ., & Schmidt, C. P. (2002). *A heuristic procedure for solving multi-plant, mutli-item, mutli-period capacitated lot-sizing* (Vol. 19; Tech. Rep. No. 1). Retrieved from https://search.proquest.com/docview/204765142/fulltextPDF/ECB0C75DF9C14B16PQ/1?accountid=27128

Shang, C., & You, F. (2018, 2). Distributionally robust optimization for planning and scheduling under uncertainty. *Computers & Chemical Engineering*, *110*, 53–68. Retrieved from https://www.sciencedirect.com/science/article/pii/S009813541730426X

Shapiro, J. F. (1993). Mathematical programming models and methods for production planning and scheduling. *Handbooks in Operations Research and Management Science*, *4*(C), 371–443.

Silver, E. A., & Meal, H. C. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, *14*(2), 64–74.

Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling*. New York: Wiley.

Spitter, J. M. (2005). *Rolling Schedule Approaches for Supply Chain Operations Planning* (Doctoral dissertation, Eindhoven University of Technology).

Stadtler, H. (1996, 11). Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lotsizing. *European Journal of Operational Research*, *94*(3), 561–581. Retrieved from https://www.sciencedirect.com/science/article/pii/0377221795000941

Stadtler, H. (1997, 6). Reformulations of the shortest route model for dynamic multi-item multi-level capacitated lotsizing. *OR Spektrum*, *19*(2), 87–96. Retrieved from http://link.springer.com/10.1007/BF01545506

Stadtler, H. (2003, 6). Multilevel Lot Sizing with Setup Times and Multiple Constrained

Resources: Internally Rolling Schedules with Lot-Sizing Windows. *Operations Research*, *51* (3), 487–502. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.51.3.487.14949

Suerie, C., & Stadtler, H. (2003, 8). The Capacitated Lot-Sizing Problem with Linked Lot Sizes. *Management Science*, *49* (8), 1039–1054. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/mnsc.49.8.1039.16406

Taş, D., Gendreau, M., Jabali, O., & Jans, R. (2018). A capacitated lot sizing problem with stochastic setup times and overtime. *European Journal of Operational Research*, *0*, 1–14. Retrieved from https://doi.org/10.1016/j.ejor.2018.07.032

Tempelmeier, H. (2001). Supply chain planning with advanced planning systems. *Proceedings of the 3rd Aegean International Conference on Design and Analysis of Manufacturing Systems*, 19–22.

Tempelmeier, H., & Derstroff, M. (1996, 5). A Lagrangean-Based Heuristic for Dynamic Multilevel Multiitem Constrained Lotsizing with Setup Times. *Management Science*, *42* (5), 738–757. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/mnsc.42.5.738

Tempelmeier, H., & Helber, S. (1994, 6). A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures. *European Journal of Operational Research*, *75* (2), 296–311. Retrieved from https://www.sciencedirect.com/science/article/pii/0377221794900760

Thizy, J. M., & Van Wassenhove, L. N. (1985). Lagrangean relaxation for the multi-item capacitated lot-sizing problem: a heuristic implementation. *IIE Transactions*, *17* (4), 308–313.

Van Aken, J. E., Berends, H., & Van der Bij, H. (2007). *Problem-solving in Organizations*. Cambridge: Cambridge University Press. Retrieved from http://ebooks.cambridge.org/ref/id/CBO9780511618413

Van den Akker, J. M., Bouman, P. C., Hoogeveen, J. A., & Tönissen, D. D. (2016). Decomposition approaches for recoverable robust optimization problems. *European Journal of Operational Research*, *251* (3), 739–750.

Van Donselaar, K. H., De Kok, A. G., & Rutten, W. (1996). Two replenishment strategies for the lost sales inventory model: A comparison. *International Journal of Production Economics*.

Wagelmans, A., Van Hoesel, S., & Kolen, A. (1992). Economic lotsizing: an O(n log n) algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, *40*, 145–155.

Wagner, H. M., & Whitin, T. M. (1958). A dynamic version of the economic lot size model. *Management Science*, *5* (1), 89–96.

Yanıkoglu, I., Gorissen, B. L., & Den Hertog, D. (2018, 9). A survey of adjustable robust optimization. *European Journal of Operational Research*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221718307264

Appendix A

Truncated Normal Distribution

When applying the SAA method a scenario tree had to be generated. We assumed a truncated normal distribution when generating demand for the tree. However, we need a method to generate the first two moments of the underlying normal distribution. Hence, we look at the stochastic variable $Y = \max\{0, X\}$, where X has a normal distribution with mean μ and standard deviation σ. This truncation approach shifts all probability mass from the negative support to a point mass in the origin. The remainder of this section will derive the first two moments of Y as a function of the two dispersion measures of X. $\phi(x)$ and $\Phi(x)$ are the standard normal pdf and cdf, respectively. We let $X \sim N(\mu, \sigma)$. Hence, the pdf of X is given by $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

$$
\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[\max\{0, X\}] = \int_{-\infty}^{0} 0 \cdot f(x)dx + \int_{0}^{\infty} xf(x)dx = \int_{0}^{\infty} \sigma \frac{(x-\mu+\mu)}{\sigma} f(x)dx \\
&= \int_{0}^{\infty} \sigma \frac{(x-\mu+\mu)}{\sigma} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \int_{\frac{0-\mu}{\sigma}}^{\infty} \sigma(u + \frac{\mu}{\sigma}) \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \\
&= \sigma \int_{\frac{0-\mu}{\sigma}}^{\infty} u\phi(u)du + \mu \int_{\frac{0-\mu}{\sigma}}^{\infty} \phi(u)du = \sigma \int_{\frac{0-\mu}{\sigma}}^{\infty} -\phi'(u)du + \mu(1 - \Phi(\frac{-\mu}{\sigma})) \\
&= \sigma\phi(\frac{-\mu}{\sigma}) + \mu(1 - \Phi(\frac{-\mu}{\sigma})).
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\mathbb{E}[Y^2] &= \mathbb{E}[(\max\{0, X\})^2] = \int_{-\infty}^{0} 0 \cdot f(x)dx + \int_{0}^{\infty} x^2 f(x)dx = \int_{0}^{\infty} \sigma^2 (\frac{(x-\mu+\mu)}{\sigma})^2 f(x)dx \\
&= \int_{0}^{\infty} \sigma^2 (\frac{(x-\mu+\mu)}{\sigma})^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \int_{\frac{0-\mu}{\sigma}}^{\infty} \sigma^2 (u + \frac{\mu}{\sigma})^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \\
&= \sigma^2 \int_{\frac{0-\mu}{\sigma}}^{\infty} u^2\phi(u)du + 2\mu\sigma \int_{\frac{0-\mu}{\sigma}}^{\infty} u\phi(u)du + \mu^2 \int_{\frac{0-\mu}{\sigma}}^{\infty} \phi(u)du = \\
&= \sigma\mu\phi(\frac{-\mu}{\sigma}) + (\mu + \sigma)^2(1 - \Phi(\frac{-\mu}{\sigma})),
\end{aligned}
$$

where the last equation follows from $\int_a^b u^2\phi(u)du = a\phi(a) - b\phi(b) + \Phi(b) - \Phi(a)$. These formulas can be utilized in a root finding procedure with input $\mathbb{E}[Y]$ and $\mathbb{E}[Y^2]$ to find the matching μ and σ.

Appendix B

Mean Absolute Deviation Gamma Distribution

In this note, the exact formula for the MAD of a gamma distribution is given as a function of the scale α and rate β parameters. The r.v. $X \sim Gamma(\alpha, \beta)$ models the demand uncertainty of our general model. Note that for gamma distributions the following relationship holds:

$$\tilde{d} = 2\sigma^2 f(\mu). \tag{B.1}$$

In other words, the mean absolute deviation is twice the variance σ^2 multiplied with the pdf value at the mean μ. This results can be derived in the following way. First of all, let us state

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1} e^{-x\beta}}{G(\alpha, \beta)} \tag{B.2}$$

$$G(\alpha, \beta) = \int_0^\infty u^{\alpha-1} e^{-u\beta} du = \frac{1}{\beta^\alpha}\Gamma(\alpha). \tag{B.3}$$

Furthermore, it can be shown that

$$\mu = \frac{d}{d\beta}(\ln(G(\alpha, \beta))), \quad \sigma^2 = -\frac{d\mu}{d\beta}, \tag{B.4}$$

$$\frac{df}{d\beta} = \frac{\int_0^\infty u^{\alpha-1} e^{-u\beta} du \cdot -x^{\alpha-1} x e^{-x\beta} + \int_0^\infty u^{\alpha} e^{-u\beta} du \cdot x^{\alpha-1} e^{-x\beta}}{(G(\alpha, \beta))^2} = (\mu - x) f(x; \alpha, \beta), \tag{B.5}$$

$$\text{and} \quad \int_0^\infty (x-\mu) f(x) dx = -\int_0^\mu (\mu - x) f(x) dx + \int_\mu^\infty (x-\mu) f(x) dx = 0 \tag{B.6}$$

Hence, it is easy to derive formula (B.1) by noticing that

$$\tilde{d} = \int_0^\infty |x-\mu| f(x; \alpha, \beta) dx \overset{(B.6)}{=} 2\int_0^\mu (x-\mu) f(x; \alpha, \beta) dx \tag{B.7}$$

$$\overset{(B.5)}{=} -2\int_0^\mu \frac{df}{d\beta} dx = -2\frac{d}{d\beta}\int_0^\mu f(x; \alpha, \beta) dx = -2\frac{d}{d\beta}F(\mu) = -2\frac{d\mu}{d\beta} f(\mu) \overset{(B.4)}{=} 2\sigma^2 f(\mu). \tag{B.8}$$

Now it is trivial to derive the formula for the MAD of the gamma distribution by using equations (B.8) and the expressions for the dispersion measures of this distribution:

$$\tilde{d} = \frac{2e^{-\alpha}\alpha^\alpha}{\Gamma(\alpha)\beta}. \tag{B.9}$$

Appendix C

Proof Theorem 6.1.1

Proof. We consider the following model as our main problem:

$$\min \quad \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}[h_k I_{kt} + b_k L_{kt}] \tag{C.1}$$

$$s.t. \quad I_{k,t-1} + Q_{k,t-z_k} - \sum_{j\in\mathcal{S}_k} a_{kj}Q_{jt} = I_{k,t} \qquad \forall k\in\mathcal{K}\backslash\mathcal{E},\ \forall t\in\mathcal{T} \tag{C.2}$$

$$I_{e,t-1} + L_{et} + Q_{e,t-z_e} = I_{e,t} + d_{et} \qquad \forall e\in\mathcal{E},\ \forall t\in\mathcal{T} \tag{C.3}$$

$$\sum_{k\in\mathcal{K}_m} p_k Q_{kt} \le c_{mt} \qquad \forall m\in\mathcal{M}, \forall t\in\mathcal{T} \tag{C.4}$$

$$Q_{kt} \ge q_k^{min} Y_{kt} \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{C.5}$$

$$Q_{kt} \le (\sum_{m:k\in\mathcal{K}_m} c_{mt})Y_{kt} \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{C.6}$$

$$Q_{kt}, I_{kt}, L_{kt} \ge 0 \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{C.7}$$

$$Y_{kt} \in \{0,1\} \qquad \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{C.8}$$

We want to proof that this model lies in NP-hard. To this purpose, we show that the Facility Location Problem is a special case of our model. First of all, let us define $u_{k,t,t'}$ as the fraction of product k's demand produced in period t that satisfies demand in period t'. We take $\mathcal{K} = \mathcal{E}$ and $a_{kj} = 0$, $\forall k \neq j$. Hence, we reduce our multi-level problem to a single-level problem without product dependencies. We substitute $Q_{k,t} = \sum_{u=t}^{T} u_{k,t,u} d_{ku}$.
To remove all original inventory related variables from our model, we redefine the holding costs as $h^{'}_{k,t,t'} = \sum_{u=t}^{t'-1} h_k d_{kt'}$, which represent the cost of carrying inventory from period t to t' to satisfy demand. As we defined the variable $u_{k,t,t'}$ as the fraction of demand fulfilled in period t', we can use $h^{'}_{k,t,t'}$ to compute the total holding costs. Furthermore, we define $b^{'}_{kt} = b_k \cdot d_{kt}$. Finally, we take $q^{min} = 0$.

Now, we construct the following equivalent Facility Location Problem formulation:

$$\min \quad \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\left[\sum_{t'=t}^{T} h'_{k,t,t'} u_{k,t,t'} + b_k(1 - \sum_{u=1}^{t} u_{k,u,t})\right] \tag{C.9}$$

$$s.t. \quad \sum_{t=1}^{t=t'-z_k} u_{k,t,t'} \leq 1 \qquad \forall k \in \mathcal{K},\ \forall t' \geq t + z_k \tag{C.10}$$

$$u_{k,t,t'} \leq Y_{kt} \qquad \forall k \in \mathcal{K}\ \forall t \in \mathcal{T},\ \forall t' \geq t + z_k \tag{C.11}$$

$$\sum_{k\in\mathcal{K}_m} p_k\big(\sum_{u=t}^{T} u_{k,t,u} d_{ku}\big) \leq c_{mt} \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{C.12}$$

$$u_{k,t,t'} \geq 0 \qquad \forall k \in \mathcal{K},\ \forall t \in \mathcal{T},\ \forall t' \geq t + z_k \tag{C.13}$$

$$Y_{kt} \in \{0, 1\} \qquad \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \tag{C.14}$$

As this problem is NP-hard, by reduction we showed that our problem lies in the space of NP-hard problems. ■

Appendix D

Business Case

The different methods are tested on a company specialized in the development and production of excipients, which are used in the pharmaceutical industry. This case study targets a specific plant of the company. The structure of the internal supply chain is given in Figure D.1. First, suppliers provide the raw materials that are used as ingredients for the excipients. These are stored in a central stock point. Subsequently, eighteen items are produced at three different machines: the roler, spray, and agglo. Each of these machines is a resource with limited capacity. The target of this study is to generate profit whilst controlling the amount of investments stuck in inventory. As such, the SCOP model is optimized by including lost-sales opportunity

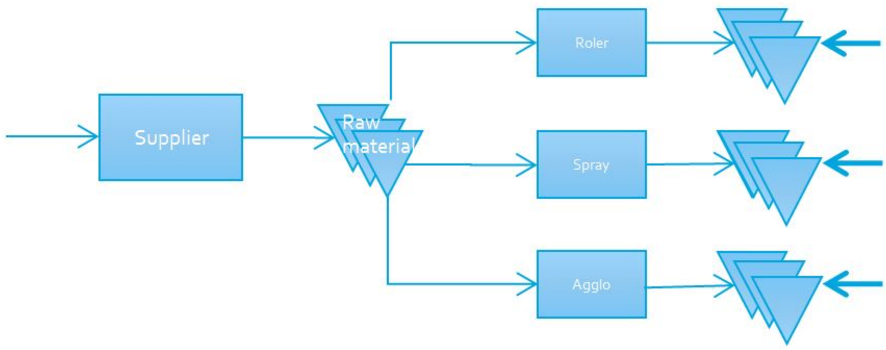
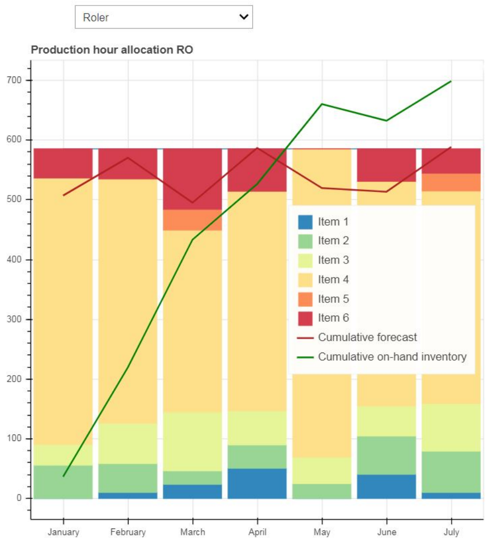
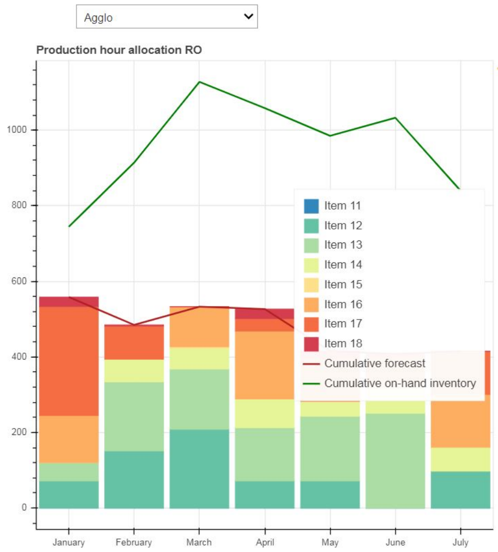


Figure D.1: Case study example

costs for unfulfilled demand and cost of capital for the inventory on hand. To solve the SCOP planning problem, we resort to the planning approaches that we discussed in the earlier sections. A visualization of the planning solutions is given in Figures D.2 and D.3. In Figure D.2a the allocation of production hours for the roler is shown under a robust optimization planning optimization. Figure D.2b shows the planning for the agglo machine. Notice that the RO approach utilizes all available capacity even though some of this could result in abundant inventory. Furthermore, the RO method is selective on the items it produces at the agglo. The capacity for this machine equals the forecasted amount of production hours. Therefore, building up additional safety stocks is restricted. However, surplus inventory is already present for the items that are produced on this machine. Figure D.3a displays the allocation of production capacity when using the SAA planning optimization methodology. This method is less conservative, and as such it uses less of the available resource. This reduces inventory cost of capital in exchange for (possibly) deteriorating service levels and increased opportunity costs due to lost profits. The DRO approach, which is displayed in Figure D.3b, positions itself in the middle ground. Only at the end of the planning horizon DRO decides to arrange leftover capacity, because cumulative inventory on hand is considered sufficient.

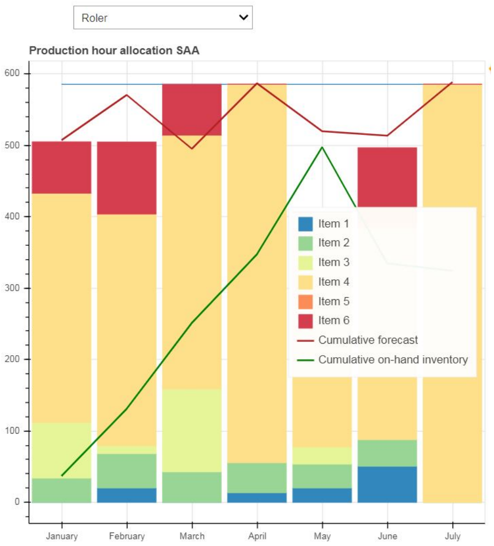


(a) Utilization roler machine with RO planning policy

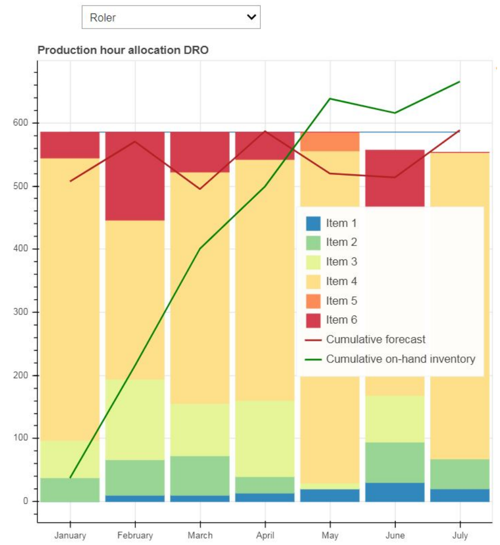


(b) Utilization agglo machine with RO planning policy

Figure D.2: Robust optimization planning optimization



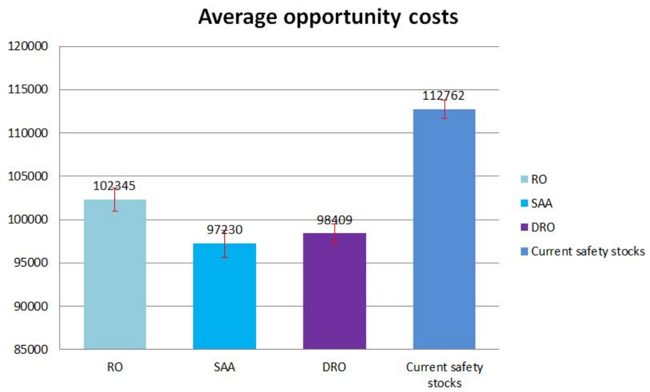
(a) Utilization roler machine with SAA planning policy



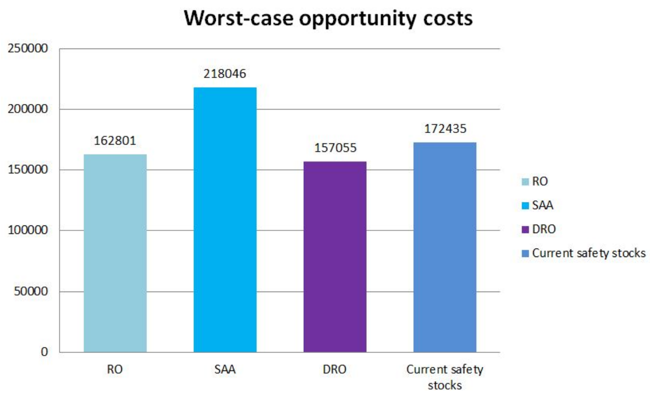
(b) Utilization roler machine with DRO planning policy

Figure D.3: SAA and DRO planning optimization

To enable comparison of our methods, an extensive simulation study was executed. Furthermore, we also studied the performance of the plant when their own predetermined safety stock levels were implemented. This enables us to quantify the relative improvement that our methods would provide compared to the current situation. The study focused on minimizing the sum of opportunity costs (lost profits due to unsatisfied customer demand) and inventory cost of capital.
Figure D.4 provides us with the results. It turns out that the SAA approach beats all other approaches when it comes down to minimizing the expected value of the aforementioned opportunity costs. SAA provides a relative improvement of approximately 14%. RO and DRO provide relative improvements close to 9% and 13% percent, respectively. However, notice that the latter methods present a more adequate protection against the worst-case scenario. Sample average approximation is severely outperformed in this area, which is due to its reduced inventory buffers.



(a) Average opportunity costs for each planning approach



(b) Worst-case opportunity costs for each planning approach

Figure D.4: Case study results

Finally, Figure D.5 gives a more detailed overview of the costs incurred by the different planning approaches. Notice that both robust optimization methods are able to generate more profit with less capital stuck in inventory. SAA reduces the cost of capital even further while only marginally reducing customer sales potential.

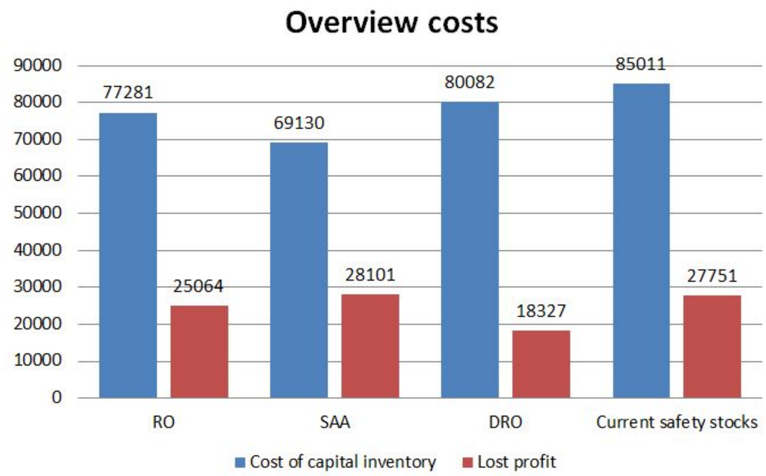


Figure D.5: Overview inventory capital and lost-sales opportunity costs for each planning approach